DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SUPERSONIC AIRCRAFT ATTACK LOSS STATUS
USING SEQUENTIAL SIMPLEX MATERIAL
FOR FIXED ELEMENT LOSS

by

David W. Davidson

September 1969

Thesis Advisor: Arthur R. Tyler

Approved for public release; distribution unlimited.

Prepared for: Defense Communications Agency
1860 Wiehle Avenue
Reston, Va. 22090

T22060 T222052

NPS-67-84-016

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

SUPERSONIC MISSILE SEEKER LENS SYSTEM
USING GRADIENT INDEX MATERIAL
FOR FIRST ELEMENT LENS

by

David S. Davidson

September 1984

Thesis Advisor: Allen E. Fuhs

Approved for public release; distribution unlimited.

Prepared for: Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

T222057 T222058

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Commodore R. H. Schumaker
Superintendent

David A. Schrady
Provost

This thesis prepared in conjunction with research supported in part by Defense Advanced Research Projects Agency under DARPA Order 4035.

Reproduction of all or part of this report is authorized.

Released as a
Technical Report by:

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NPS-67-84-016
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): Supersonic Missile Seeker Lens System Using Gradient Index Material for First Element Lens
5. TYPE OF REPORT & PERIOD COVERED: Master's Thesis; September 1984
6. PERFORMING ORG. REPORT NUMBER: NPS-67-84-016

7. AUTHOR(s): David S. Davidson
8. CONTRACT OR GRANT NUMBER(s): N622718WE40007

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93943
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:

11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93943
12. REPORT DATE: September 1984
13. NUMBER OF PAGES: 119

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): COL Rene Larriva, USMC, Defense Advanced Research Prljects Agency, 1400 Wilson Boulevard, Arlington, Virginia 22209
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

18. SUPPLEMENTARY NOTES:

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
Missile Sensors
Missile Aerodynamics
Projectile Aerodynamics
Gradient Refractive Index
Lens Design
Conical Optics
Conical Lens
Guided Projectile

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):
The design of a supersonic missile seeker lens system using an aerodynamically efficient lens, a scanning mirror and detector was accomplished through the use of ray-tracing routines. The design of the first element lens uses a gradient in refractive index to overcome the severe handicap that the pointed shape imposes. This problem has been previously solved for positive orientations of the center of symmetry of the Gradient Refractive Index (GRIN) material with respect to the lens. In this study the negative

20. Abstract continued

orientations are resolved and the lens studies for a range of orientations and strengths of gradient. The best lens is then used in conjunction with the mirror and detector in a ray-tracing scheme. Such a system was found to be feasible and worthy of further study.

Approved for public release; distribution unlimited.

SUPERSONIC MISSILE SEEKER LENS SYSTEM USING GRADIENT INDEX MATERIAL FOR FIRST ELEMENT LENS

by

David S. Davidson
Captain, Canadian Forces
B. Eng (Engineering Physics), Royal Military College, 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN PHYSICS

and

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 1984

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

ABSTRACT

The design of a supersonic missile seeker lens system using an aerodynamically efficient lens, a scanning mirror and detector was accomplished through the use of ray-tracing routines. The design of the first element lens uses a gradient in refractive index to overcome the severe handicap that the pointed shape imposes. This problem has been previously solved for positive orientations of the center of symmetry of the Gradient Refractive Index (GRIN) material with respect to the lens. In this study the negative orientations are resolved and the lens studies for a range of orientations and strengths of gradient. The best lens is then used in conjunction with the mirror and detector in a ray-tracing scheme. Such a system was found to be feasible and worthy of further study.

TABLE OF CONTENTS

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

I. INTRODUCTION 15

A. OUTLINE OF LENS DESIGN TO DATE 15

B. GRADIENT REFRACTIVE INDEX (GRIN) 15

C. AIM AND INTENT OF THIS STUDY 16

II. GRIN LENS DESIGN 18

A. GRIN LENS DESIGN METHODOLOGY 18

B. GRIN RAY PATH DESCRIPTION 19

C. GRIN LENS GEOMETRY 22

D. RAY MATCHING 24

III. NEAR-RADIAL LINES 30

A. DESCRIPTION OF DIFFICULTY 30

B. PRELIMINARY WORK - IMPROVING THE ITERATION SCHEME 31

C. PRELIMINARY WORK - DOUBLE PRECISION 36

D. PRELIMINARY WORK - CHECKING ACCURACY 36

E. FIRST PROPOSED SOLUTION - STRAIGHT LINES . . .38

F. SECOND PROPOSED SOLUTION - REDUCED EQUATION .38

G. THIRD PROPOSED SOLUTION - MARCHAND METHOD . .45

H. FOURTH PROPOSED SOLUTION - ANGLE IN TERMS OF RADIUS 46

IV. GRADIENT INDEX SEEKER LENS RAY TRACE RESULTS . . .53

A. FRONT SURFACE ACCURACY 53

B. SPOT SIZE RESULTS 53

C. SELECTION OF A BEST LENS 53

V. MIRROR AND DETECTOR SYSTEM 57

A. DESCRIPTION OF SYSTEM 57

B. LENS TO DETECTOR RAY TRACE 58

1. Ray Direction Exiting Lens 58

2. Point of Intersection on Mirror 58

3. Ray Direction Exiting Mirror 61

4. Ray Count on Detector 64

C. MIRROR AND DETECTOR SYSTEM RESULTS 65

VI. CONCLUSIONS AND RECOMMENDATIONS 68

A. CONCLUSIONS 68

B. RECOMMENDATIONS 69

APPENDIX A: PROGRAM GISL LISTING 70

APPENDIX B: PROGRAM CHECK LISTING 98

APPENDIX C: PROGRAM DETPLOT LISTING 103

APPENDIX D: DETECTOR SIGNAL 107

LIST OF REFERENCES 117

INITIAL DISTRIBUTION LIST 118

LIST OF TABLES

I. Front Surface Accuracy 55

LIST OF FIGURES

2.1 GRIN Lens Coordinate System and Parameters (from Carr) 20

2.2 GRIN Lens and Ray Geometry 23

2.3 Ray Path Showing Improper Matching and Minimum Radius 26

2.4 Ray Path Radius and Angle 27

2.5 Ray Matching Regions 29

3.1 Ray Trace Showing Region of Radial Lines 32

3.2 Newton-Raphson Iteration Geometry 34

3.3 Newton-Raphson Logic Flowchart 35

3.4 Geometry of Accuracy Check - Forward Ray Trace .39

3.5 Expanded View of Front Surface - Straignt Line Method 41

3.6 Reduced Equation Implementation 44

3.7 Radius and Angle Relation 48

3.8 Angle in Terms of Radius Iteration Logic Flowchart 50

3.9 Angle in Terms of Radius Iteration Geometry . . .51

3.10 Diverging Iteration - Forward Ray Trace 52

4.1 Spot Size Results - ALFAP = 0.3 Radians 56

5.1 Lens, Mirror and Detector System 59

5.2 Detector Signal 60

5.3 Reflection of Ray on Mirror 63

5.4 Mirror Tilt for On-Axis Spot and Angle of Incidence 67

LIST OF SYMBOLS

SYMBOL	FORTRAN EQUIVALENT	DEFINITION	UNITS
a	A	index function parameter	nondimensional
arg	ARG	argument of arcsine function in GRIN ray equation	nondimensional
arg_0	-	argument of initial arcsine function in GRIN ray equation	nondimensional
b	B	index function parameter	nondimensional
c	C	ray matching constant	radians
$D_{\ell m}$	D1	ray geometrical distance from lens to mirror	nondimensional
D_{nd}	D2	ray geometrical distance from mirror to detector	nondimensional
DT	DETP	position of detector along x-axis	nondimensional
e	E	GRIN ray trace constant	nondimensional
K	CK	x-direction cosine of ray from lens to mirror	nondimensional
K'	CKK	x-direction cosine of ray from mirror to detector	nondimensional
ℓ_0	-	z-direction cosine of ray at initial point	nondimensional
L	CL	y-direction cosine of ray from lens to mirror	nondimensional
L'	CLL	y-direction cosine of ray from mirror to detector	nondimensional

M	CM	z-direction cosine of ray from lens to mirror	nondimensional
M'	CMM	z-direction cosine of ray from mirror to detector	nondimensional
M_0	-	Marchand reduced integral expression	-
MP	MIRP	position of mirror pivot point on x-axis	nondimensional
n	-	index of refraction	nondimensional
n_0	-	index of refraction at initial point of GRIN ray	nondimensional
$\vec{n}$		vector normal to mirror plane	-
n_x	-	x-component of vector normal to mirror plane	nondimensional
n_y	-	y-component of vector normal to mirror plane	nondimensional
p_0	-	x-direction cosine of GRIN ray at initial point	nondimensional
q_0	-	y-direction cosine of GRIN ray at initial point	nondimensional
r	RAD	radius to ray from center of symmetry	nondimensional
r_0	RO	radius to ray at initial point	nondimensional
r'	RP	first guess radius in iteration scheme	nondimensional
r"	-	second guess radius in iteration scheme	nondimensional
r"'	-	third guess radius in iteration scheme	nondimensional
r_{fs}	-	radius to front surface	nondimensional

$\dot{r}_{fs}$	-	derivative with respect to angle of radius to front surface	nondimensional
r_{gr}	-	radius to GRIN ray	nondimensional
$\dot{r}_{gr}$	-	derivative with respect to angle of radius to GRIN ray	nondimensional
$\vec{r}_i$	-	ray vector incident to mirror	nondimensional
r_{xi}, r_{yi}, r_{zi}	-	components of ray vector incident to mirror	nondimensional
$\vec{r}_r$	-	ray vector reflected off mirror	-
r_{xr}, r_{yr}, r_{zr}	-	components of ray reflected off mirror	nondimensional
R	R	radius from x-axis to outermost point of inner surface	nondimensional
R_z	Rzero	radius to edge of GRIN material	nondimensional
U	U	angle of incident radiation for lens construction	radians
x_d	XD	x-component of ray at detector	nondimensional
x_m	XM	x-component of ray at mirror	nondimensional
x_0	XO	x-component of ray at lens inner surface	nondimensional
y_d	YD	y-component of ray at detector	nondimensional
y_m	YM	y-component of ray at mirror	nondimensional

y_0	YO	y-component of ray at lens inner surface	nondimensional
z_d	ZD	z-component of ray at detector	nondimensional
z_m	ZM	z-component of ray at mirror	nondimensional
z_0	ZO	z-component of ray lens inner surface	nondimensional
ε	EPSLON	sign function of GRIN ray equation	-
ψ	PSI	angle between GRIN ray and radial arm from center of symmetry	radians
ψ_0	PSIO	angle between GRIN ray and radial arm at initial point	radians
θ	THETA	angle from x-axis to radius	radians
θ_0	THETO	angle from x-axis to initial point	radians
θ_L	-	angle for GRIN ray part to left of part of minimum radius	radians
θ_R	-	angle for GRIN ray part to right of point of minimum radius	radians
θ'	THETAP	angle of first guess in iteration scheme	radians
θ''	-	angle to sedonc guess in iteration scheme	radians
θ'''	-	angle of third guess in iteration scheme	radians
θ_m	THETM	angle from x-axis to plane of mirror	radians

ACKNOWLEDGEMENTS

The author would like to thank Distinguished Professor Allen E. Fuhs for his guidance and perserverance during the writing of this thesis.

The author would also like to express his gratitude to his wife Johanne for her support during this period.

I. INTRODUCTION

A. OUTLINE OF LENS DESIGN TO DATE

The design of a 'pointed' aerodynamically efficient missile seeker lens has been the topic of a number of studies. Frazier [Ref. 1], and Terrell [Ref. 2], first studied the idea of using a pointed lens as a means of obtaining some degree of aerodynamic efficiency. They also introduced the idea of using Gradient Refractive Index (GRIN) material in order to overcome the difficulties in imaging imposed by the odd shape. Amichai [Ref. 3], and Carr [Ref. 4], studied the GRIN concept in more detail, Amichai by defining the outer surface as a cone and solving for the inner surface and Carr by defining the inner surface as a cone and solving for the outer. The main reference for this author's study was drawn from Carr's work and a review of that thesis would be helpful if the reader seeks a more detailed background from that given in the next few pages.

B. GRADIENT REFRACTIVE INDEX (GRIN)

Although the theory for GRIN has been developed and known for some time it was not until fairly recently that materials have become available that allow the theory to be put into practice. A gradient in index will cause light rays to bend in an otherwise homogeneous material. Thus,

the lens designer now has a new dimension of flexibility which will allow him to solve problems that could not be formerly solved. A proper selection of the manner in which the gradient varies is also important. The designer would normally select certain degrees of symmetry. Three are most common; axial, radial or spherical gradients.

GRIN material in the shape of rods now commonly use axial and radial gradients in such devices as fiber-optic cables and photocopying machines. Spherical gradients are not available because spherical gradient material has not yet been produced. Spherical gradients do offer an advantage in that they lend themselves more easily to mathematical description since closed form solutions for the ray paths can be obtained.

A good introduction to gradients and the materials currently available for production of large scale GRIN lens systems is provided by Moore [Ref. 6], and Light [Ref. 7]. These also give the reader an idea of the advantages that may be realized along with the problems associated with the design and production of such items.

C. AIM AND INTENT OF THIS STUDY

The aim of this study is to complete the design of the Gradient Index Seeker Lens (GISL) initiated by Carr. This design uses a spherical gradient and 'pointed' lens shape. This study does not determine the aerodynamic efficiency of

the design but rather ensures that the lens has a generally pointed character that will also allow some degree of imaging for off-axis objects. The spherical gradient is chosen since it seems to fit the geometrical shape of the lens and also since the three-dimensional analysis of rays is much simpler.

This design is accomplished through the use of ray tracing techniques. Thus, no actual study of the lens material nor the manner in which such a lens could be manufactured is intended. It is the hope of the author that studies of this nature will provide the impetus for the development of suitable GRIN material with spherical gradients.

Once the lens design was completed a 'best' lens was chosen and a simple detector system applied to it to see if precise definition of off-axis targets is possible.

II. GRIN LENS DESIGN

A. GRIN LENS DESIGN METHODOLOGY

The method by which the GRIN Lens was designed in Carr's thesis [Ref. 4], is the subject of this section. A cross-section view of the top half of the lens is shown in Figure 2.1.

As can be seen, the lens is designed by defining the inner surface as a cone and then constructing the outer surface so that a backwards ray trace from a point source will result in parallel lines emerging. This is done by tracing rays from the back edge of the lens towards the front surface and finding the point of intersection of the ray and front surface line. The front surface line is found by using the previous ray solution as described below. The intersection of the ray and the line is determined using an iteration procedure. Once a solution is found the front surface line that will cause the ray to emerge parallel with all other rays is determined. This front surface line is then used to find a point solution for the next ray. About 1000 rays are traced starting at the outside edge of the inner cone and working inwards towards the center. Details of the iteration process is included in later sections.

Once the lens outside surface has been constructed the lens is then studied to see how well it will image three dimensional skew rays that enter the lens at varying degrees of incidence. To accomplish this, a grid is placed in front of the lens and then rays are traced from the intersection points of the grid through the lens and onto the image plane. The grid is then tilted at successive angles and the resulting 'spots' on the image plane recorded for each case. The manner in which these spots move on the image plane and their size are the major performance characteristics of the lens.

B. GRIN RAY PATH DESCRIPTION

Following Marchand [Ref. 5], GRIN ray paths in spherical gradients are found through the solution of the appropriate line integral:

$$\phi = \phi_0 + e \int_{r_0}^{r} \frac{dr}{r(n^2 r^2 - e^2)^{1/2}} \qquad (2.1)$$

where

$$e = nr\sin\theta = n_0 r_0 \sin\theta_0 \qquad (2.2)$$

and

$$\sigma = \pm 1$$

It should be noted that the value of σ is -1.0 if the radius is decreasing with angle and +1.0 if the radius is increasing with angle.

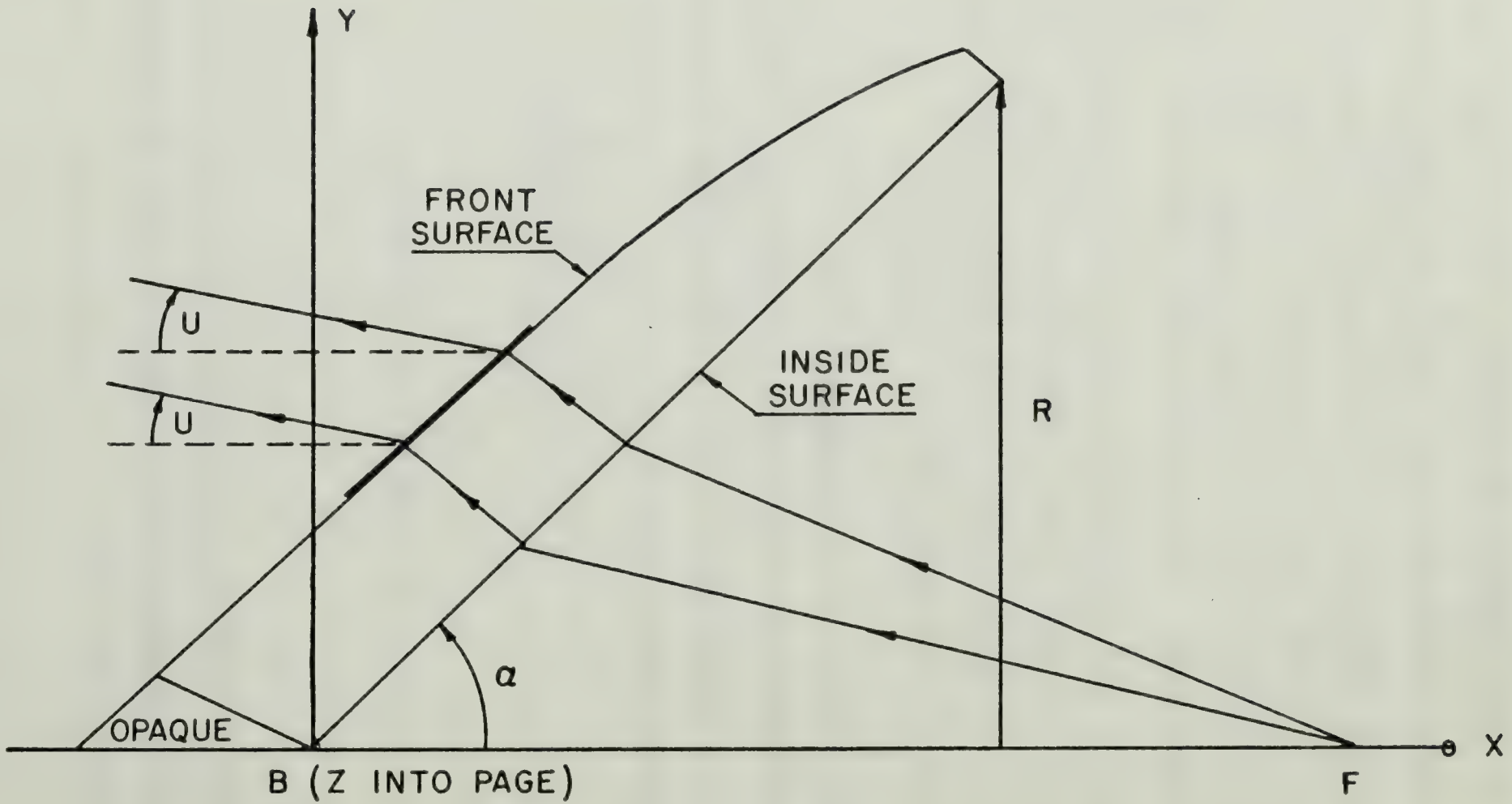

Figure 2.1 GRIN Lens Coordinate System and Parameters (from Carr).

The choice of index gradient is instrumental in allowing the integral to be solved in closed form. In this study the index is defined as:

$$n^2 = (a + br^2/R_z^2)$$

The constant a defines the strength of the index at the origin and the constant b defines the strength of the gradient and whether it is negative or positive.

As can be seen in Figure 2.2, the ray path is described by the constant e in conjunction with the parameter ψ, the angle between the radial arm of the coordinate system and the direction of the ray at any point.

The solution of the integral in equation 2.1 results in the following expression:

$$\theta = \theta_0 - \frac{\varepsilon}{2} \{\sin^{-1}(arg_0) - \sin^{-1}(arg)\} \qquad (2.3)$$

where

$$arg_0 = \frac{a - 2e/r_0^2}{(a^2 + 4be^2/R_z^2)^{1/2}}$$

and

$$arg = \frac{a - 2e/r^2}{(a^2 + 4be^2/R_z^2)^{1/2}}$$

or, expressed in terms of the radius,

$$r = \frac{\sqrt{2}|e|}{\{a+(a^2+4be^2/R_z^2)\sin[-2\varepsilon(\theta-\theta_0) - \sin^{-1}(arg_0)]\}^{1/2}} \qquad (2.4)$$

C. GRIN LENS GEOMETRY

A factor which must be considered is the manner in which the lens is cut out of the GRIN material. Figure 2.2 shows an orientation of the material with respect to the lens such that the center of symmetry of the material is inside the lens. Carr has defined this as a negative value of OB, the distance from the point O to the point B. Carr was able to work out the lens problem for positive values of this parameter only, thus the major portion of this study is to develop methods that enable solutions for negative values of OB. In so doing, two factors have to be overcome:

a) ray matching - When the ray path radius initially decreases and then increases with change in angle the ray path equation must be matched at the point of minimum radius.

b) near-radial lines - Rays that show very little curvature are difficult to describe in terms of radius and angle and these cause problems in the iterative solution scheme.

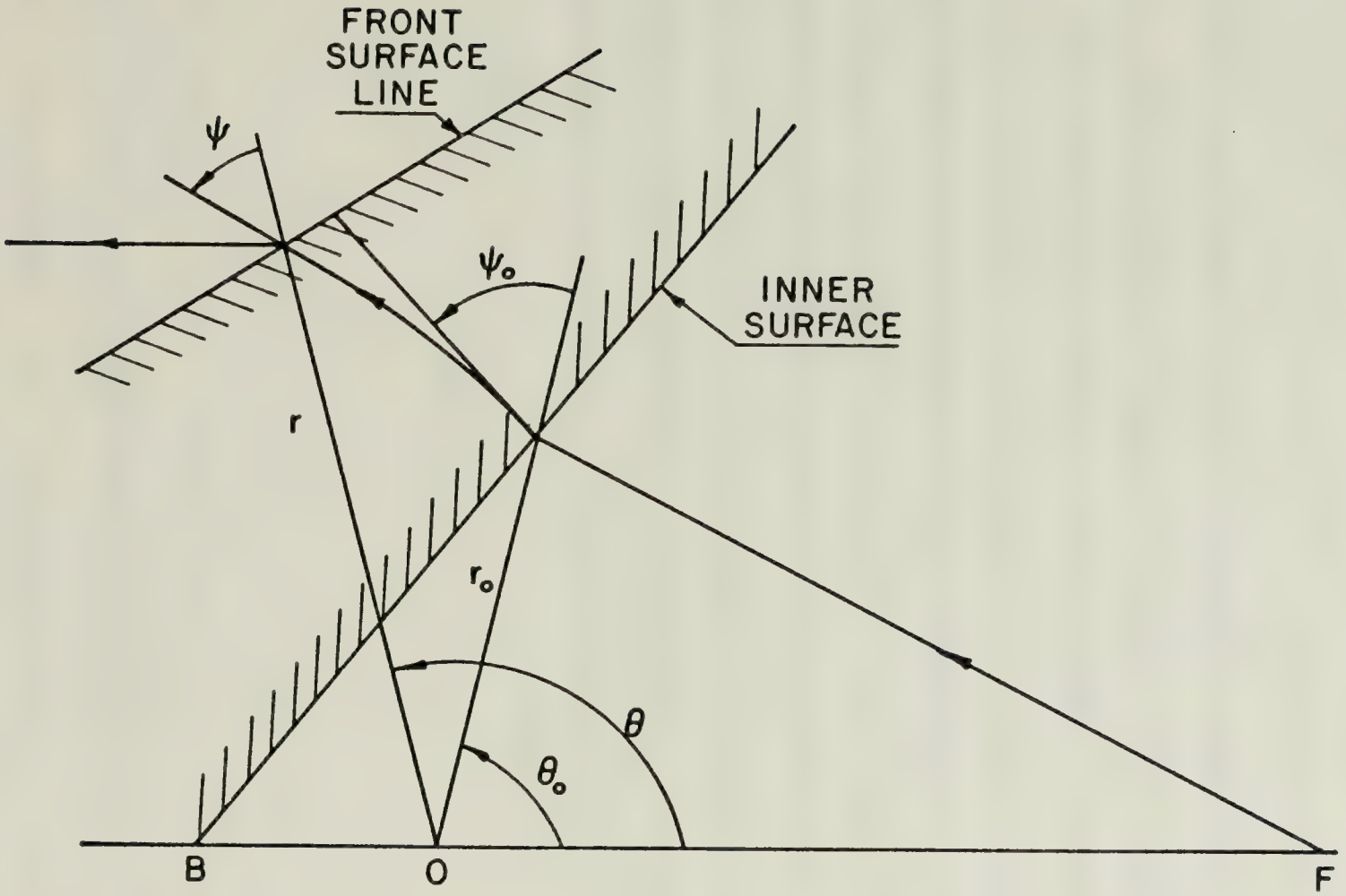

Figure 2.2 GRIN Lens and Ray Geometry.

The solution to the ray matching problem is relatively simple and is discussed in the next section. The resolution of near-radial lines is a more complex task and is covered in the following chapter.

D. RAY MATCHING

As stated previously the ray path equation changes signs as the radius of the path decreases and then increases. Such a path is shown in Figure 2.3. Simply changing sign in the equation as the point of minimum radius is passed does not produce the desired result, but rather has the effect shown in Figures 2.3 and 2.4.

The resolution of this problem is fairly simple. At the point of minimum radius the arcsine function in equation 2.3 containing this radius (arg) reduces to $-\pi/2$. This is due to the fact that at this point the value of ψ is $\pi/2$. Thus:

$$\sin \psi = 1$$

and

$$e^2 = n^2 r^2$$

Thus:

$$\arg = \frac{a-2n^2}{(a^2+4bn^2r^2/R_z^2)^{1/2}} = \frac{a-2(a+br^2/R_z^2)}{(a^2+4b(a+br^2/R_z^2)r^2/R_z^2)} = 1$$

and

$$\sin^{-1}(\text{arg}) = -\pi/2$$

If the radius is initially decreasing as shown in Figure 2.3, we can find the angle of the minimum radius. We call this the critical angle or θ_L:

$$\theta_L = \theta_0 - \frac{\varepsilon}{2}\{\sin^{-1}(\text{arg}_0) + \pi/2\}$$

The positive and the negative forms of the equation should both contain the point of minimum radius, in other words, by approaching the point from the left or the right the same result should be obtained. The fact that this does not happen is shown in Figure 2.3 and reveals the need to match the rays at the point of minimum radius.

This matching is achieved by simply adding a constant to either the positive or the negative form of equation 2.3 and then matching at the point of minimum radius in the following manner:

$$\theta_R = \theta_L + c$$

thus

$$-\frac{\varepsilon}{2}\{\pi/2+\sin^{-1}(\text{arg}_0)\} = +\frac{\varepsilon}{2}\{\pi/2+\sin^{-1}(\text{arg}_0)\} + c$$

The constant c can then be found:

$$c = -\varepsilon\{\pi/2+\sin^{-1}(\text{arg}_0)\}$$

This constant must now be applied for all points past the point of minimum radius; in our example all points in

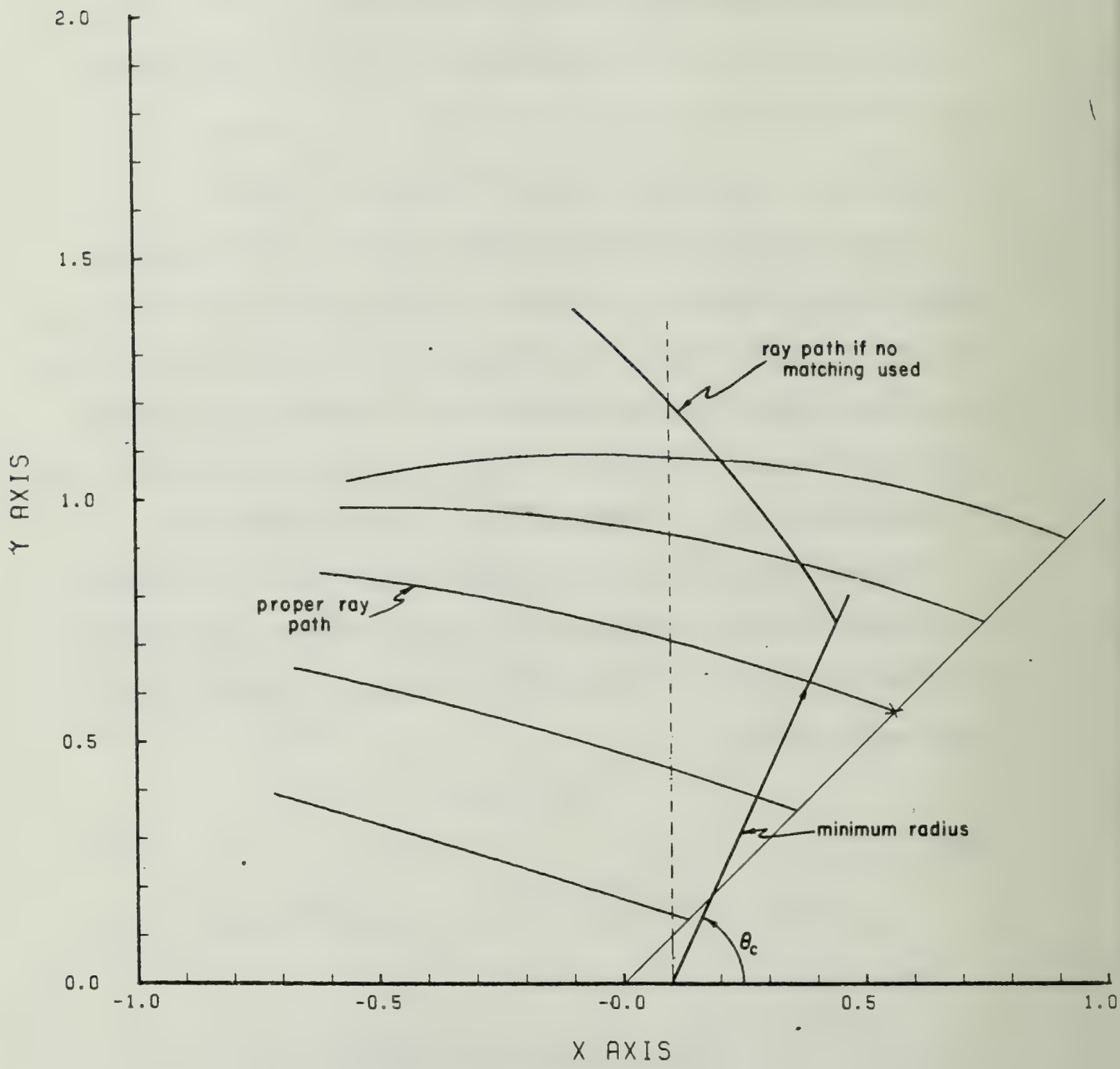

Figure 2.3 Ray Path Showing Improper Matching and Minimum Radius.

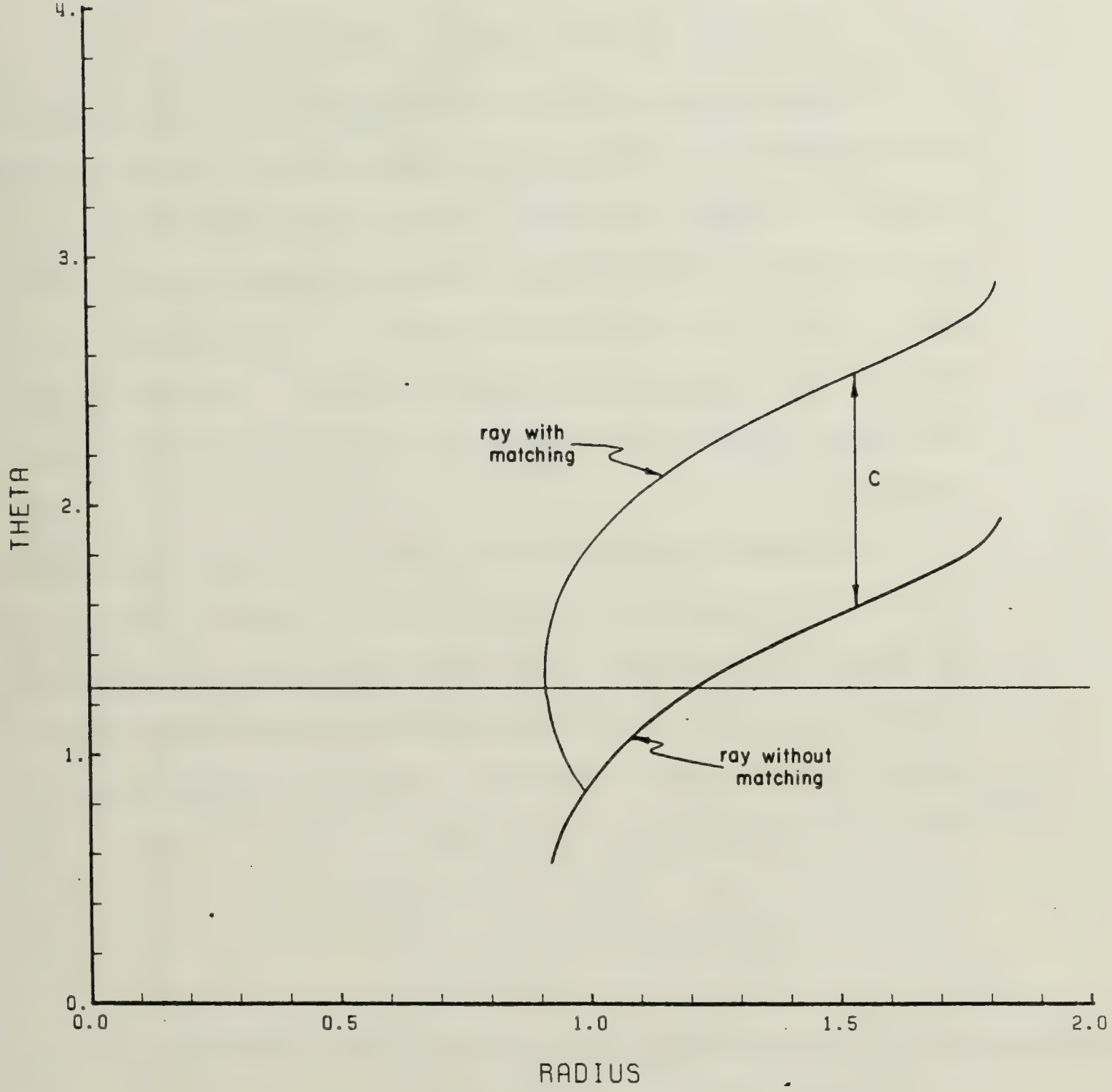

Figure 2.4 Ray Path Radius and Angle.

the increasing radius portion of the ray must be found using the form:

$$\theta = \theta_0 + \frac{\varepsilon}{2} \{\sin^{-1}(arg_0) = \sin^{-1}(arg)\} + c$$

In the iteration scheme the ray path equation must be more carefully handled. First, the critical angle and the constant c must be found. Second, the angle of the ray must be checked to see if it has passed the critical angle. If it has then the constant c is added to equation 2.3 and the point recalculated. The details of the iteration methods will be described in the following chapter.

For further clarification Figure 2.5 shows the regions in which ray matching is required for a certain position of the center of symmetry. Note also that were the value of OB positive then all rays in the lens would never reach a minimum radius point. Thus ray matching is a problem unique to the negative OB case.

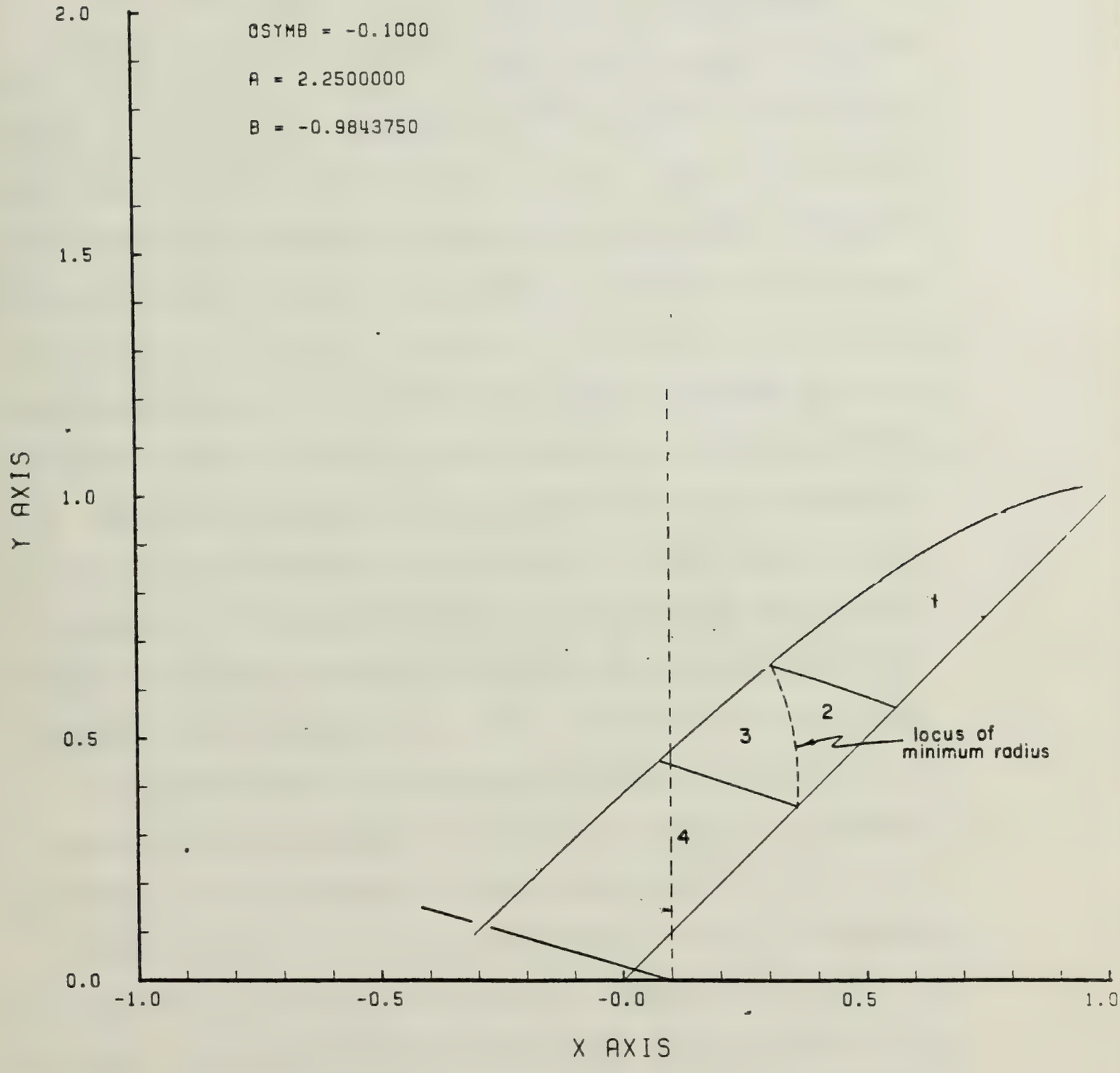

Figure 2.5 Ray Matching Regions.

III. NEAR-RADIAL LINES

A. DESCRIPTION OF DIFFICULTY

As mentioned previously, the ray paths that are nearly radial are difficult to describe mathematically in terms of radius and angle. This is due to the simple reason that a straight line originating from the center of the coordinate system has no change in angle and an undefinable change in radius.

The region in which this problem occurs in the lens design is shown in Figure 3.1. As can be seen, near-radial lines mean that the angle ψ and thus the constant e will be very small, so small as to cause equation 2.4 to approach a singularity and thus break down. In the iterative scheme, as the front surface solutions are found from the outside edge of the lens inwards, the values of ψ and e become increasingly smaller if the center of symmetry is inside the lens (OB negative). When this region is reached, equation 2.4 reduces the r_0 for all angles and futher construction of the front surface is rendered impossible. This effect is shown in Figure 3.1.

Of note is the fact that if the radial line region is passed successfully, the rays begin to curve in the opposite sense, that is they are always convex to the

radial lines in the case of a negative gradient and concave to the radial lines if a positive gradient is used.

The resolution of the near-radial lines proved to be a difficult task and became a major portion of this study. The manner in which it was overcome is the subject of the following sections.

B. PRELIMINARY WORK - IMPROVING THE ITERATION SCHEME

The first step in the resolution of the problem was to simplify the iteration scheme. In his work Carr developed a scheme based on the geometry of the rays and coordinate system [Ref. 4], which worked well for all positive values of OB (that is, for all orientations of the center of symmetry to the left of the inside surface cone. See Figure 2.2). It was decided instead to use the Newton-Raphson approach, which is simpler in concept and not dependent on geometry.

Carr used the Newton-Raphson method in the second portion of the design involving the skew ray trace, but he did not use it for the lens construction.

The manner in which this scheme works is now described so that the reader will be able to better understand the concept of how all iterative methods work for this design, as well as how the ray matching previously described fits in.

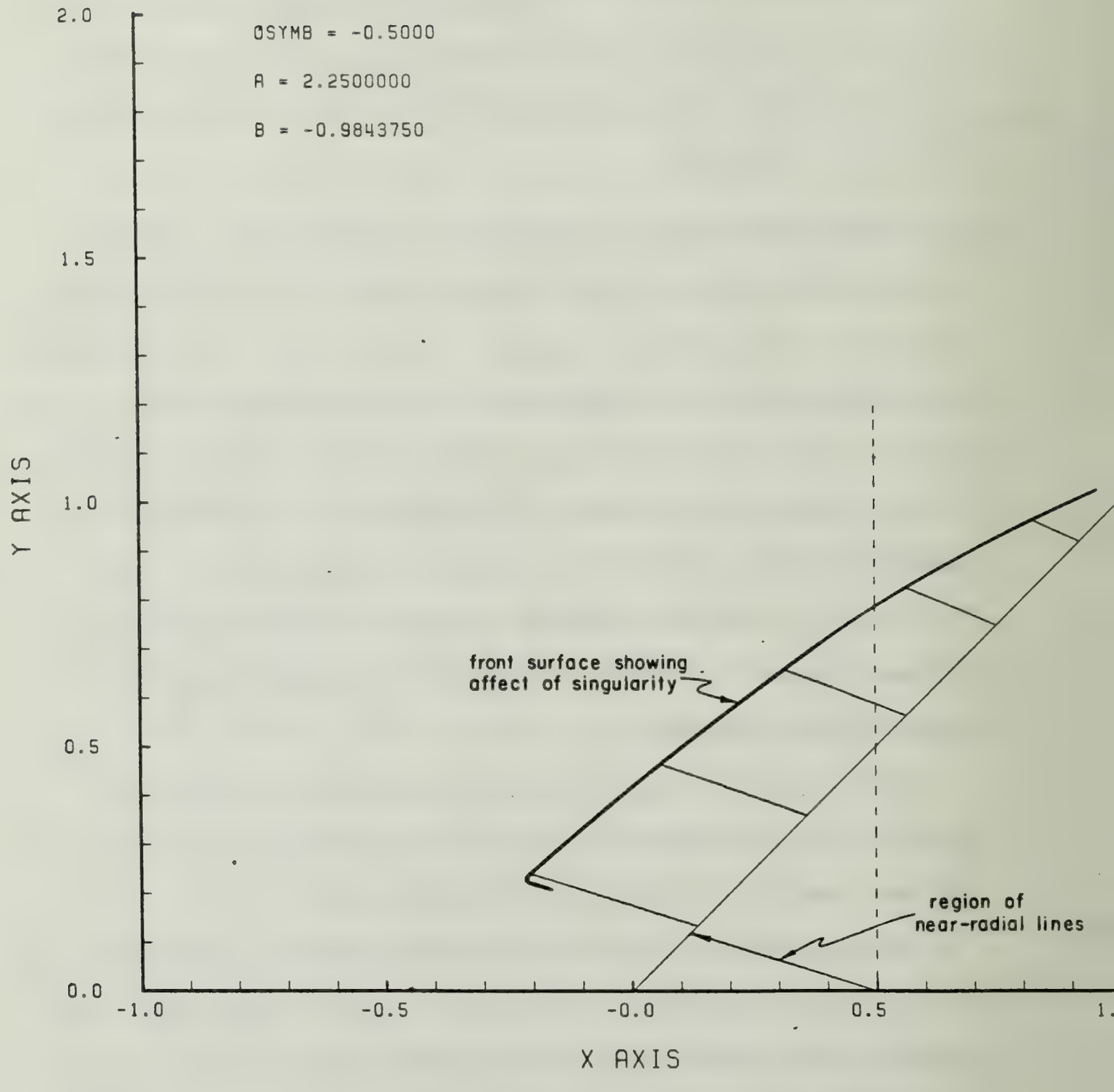

Figure 3.1 Ray Trace Showing Region of Radial Lines and Effect of Singularity.

The geometry associated with this plan is shown in Figure 3.2 and a flowchart outlining the iteration logic is shown as Figure 3.3. As can be seen with reference to the two figures, an initial point is first chosen by drawing a straight line from the inside surface intercept to the slope of the front surface obtained from a previous ray path solution. Newton-Raphson seeks to find the intercept of this front surface line with the curved ray path. This is done by using the initial angle to find the radius and derivative with respect to angle for each line and then calculating a new angle according to the relation:

$$\theta'' = \theta' - \frac{r_{fs} - r_{gr}}{\dot{r}_{fs} - \dot{r}_{gr}}$$

This new angle is then used to recalculate the radii and derivatives and thus an iterative loop if formed. Once the difference in angle is small enough the iteration is stopped and the solution considered to be found.

Ray matching is also employed in this scheme. As each new angle is calculated, it is checked to see if the critical angle has been passed. If it has, the constant c of equation 2.5 is added to the angle and thus the proper radius and derivative will be then found.

Once a solution has been found it is used to find a new front surface slope by determining that slope which will

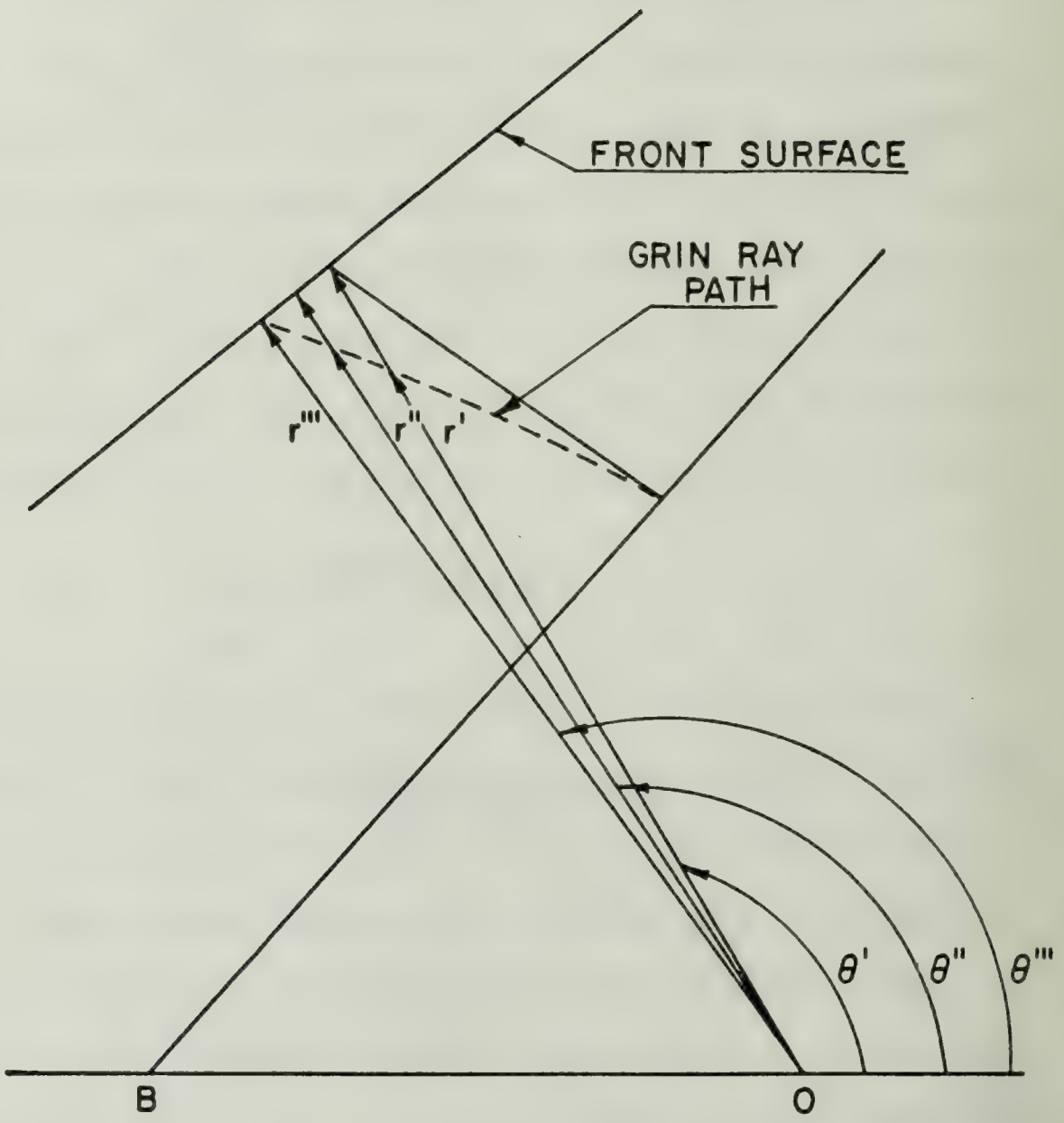

Figure 3.2 Newton-Raphson Iteration Geometry.

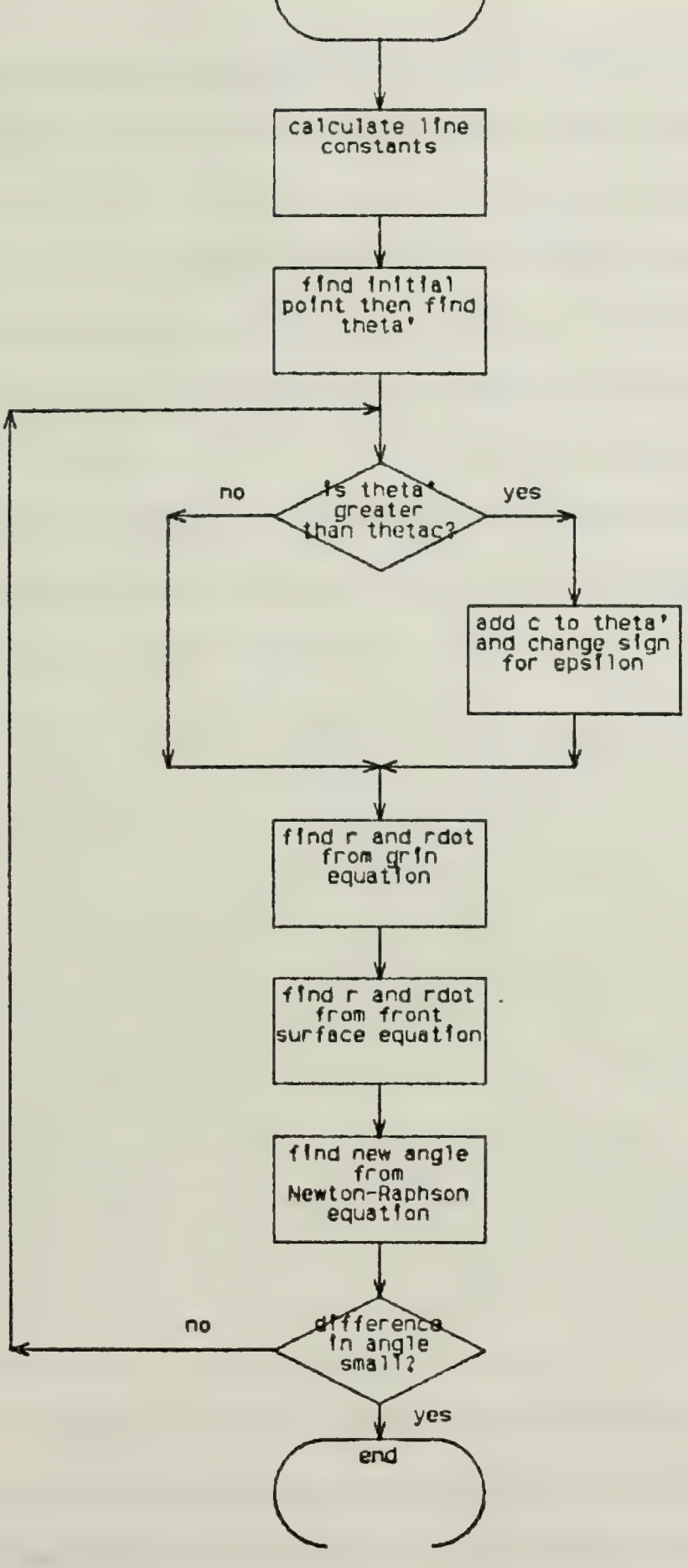
calculate line constants
find initial point then find theta'
no
Is theta' greater than thetac?
yes
add c to theta' and change sign for epsilon
find r and rdot from grin equation
find r and rdot from front surface equation
find new angle from Newton-Raphson equation
no
difference in angle small?
yes
end

Figure 3.3 Newton-Raphson Logic Flowchart.

cause the refracted ray to emerge parallel to the axis. It is this slope that is used to find the solution for the next ray in the scheme.

When this new iterative procedure was implemented it was found to work efficiently for both the positive and negative gradients, but it did not resolve the problem of near-radial lines.

The reader should refer to Appendix A for a listing of program GISL, which is the ray tracing program originally written by Carr and modified by the author.

C. PRELIMINARY WORK - DOUBLE PRECISION

The next avenue that was explored was to place the ray tracing scheme in double precision. This was considered necessary since very small differences in angle were required in the region of the radial lines.

Once this was installed in the routine it was found to improve significantly the accuracy of the calculations and succeeded in delaying the point at which the iterations broke down. It did not solve the crux of the difficulty however.

D. PRELIMINARY WORK - CHECKING ACCURACY

At this point it was decided to develop a scheme to check the accuracy of the front surface solution. The manner in which this was done is shown in Figure 3.4

The ray trace was completed in a forward direction from the front surface to the focal point in a two-dimensional fashion. These rays were caused to originate from the mid point of each set of points that define the front surface. The slope of the front surface for each of these rays was taken to be the line drawn between the two adjacent points on the front surface. In this fashion the worst case was chosen since the skew ray trace used to determine the performance of the lens uses incident rays that can hit the front surface at any point.

These rays were then traced through the lens and back towards the focal point. The x-coordinate of the focal point was defined and the y-coordinate calculated from the ray trace to see how close it would come to the y-coordinate of the focal point originally used to start construction of the front surface. A listing of this routine is shown as Appendix B.

Another benefit of this two-dimensional ray trace was that it allowed the author to confirm the methods by which a forwards ray trace is done. In this manner ray matching and sign convention in angles was checked. Also this ray trace is different in that a solution between the ray path in the lens and the back surface, rather than the front surface, was now required in the iterative solution scheme and this could now be verified to ensure that it was properly done.

Another method of checking the accuracy of the solution would be to study the front surface smoothness in the region of near-radial lines. Although not as precise as the above method this idea allowed the author to confirm that the front surface was sufficiently smooth. By this means it was also possible to see a physical representation of the lens in the region of the singularity.

E. FIRST PROPOSED SOLUTION - STRAIGHT LINES

The first proposed method of solving the mathematical difficulty was to simply replace the near-radial lines with completely straight lines just before the singularity was reached. This could be done by following the decrease of the parameter ψ_0 until it became small enough to ensure that the lines were nearly straight.

This idea was implemented but it was found not to be sufficiently accurate. It must be emphasized that an inaccuracy in even a single point could not be tolerated since this would mean that all succeeding points in the iterative solution scheme would carry that error as well.

For these reasons the straight line solution was rejected and other means of solving the difficulty were sought.

F. SECOND PROPOSED SOLUTION - REDUCED EQUATIONS

The second method that was used to resolve the problem of near-radial lines was to reduce equation 2.4 by

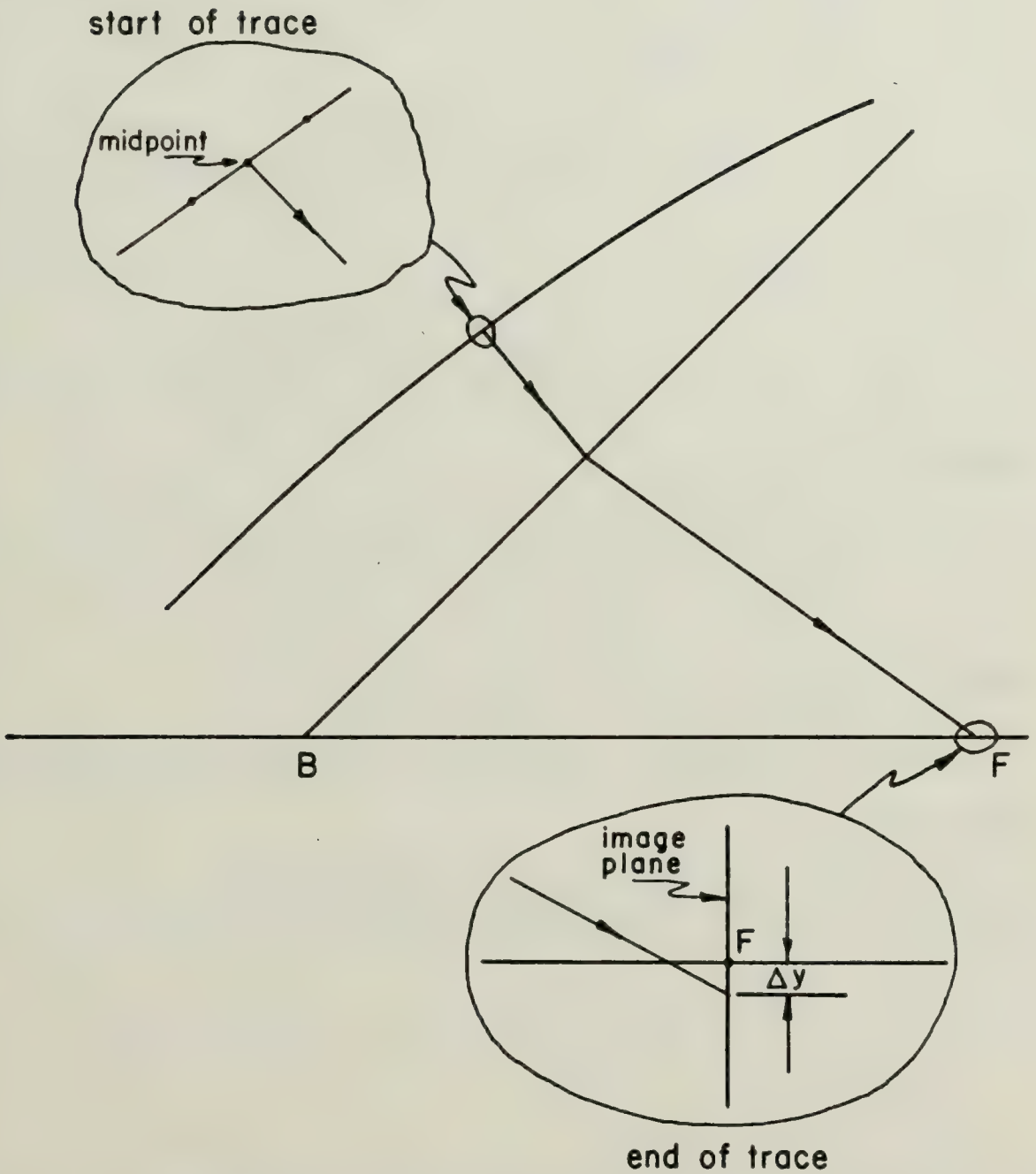

Figure 3.4 Geometry of Accuracy Check - Forwards Ray Trace.

employing small angle aproximations, binomial expansions and curve fitting. The details of how this was done follows.

Recalling equation 2.3:

$$\theta = \theta_0 - \frac{\varepsilon}{2}\left\{\sin^{-1}(arg_0) - \sin^{-1}(arg)\right\}$$

The arguments of these arcsine functions were reworked as follows:

$$arg_0 = \frac{a-2(a+b')\psi_0^2}{(a^2+4b'(a+b')\psi_0^2)^{1/2}}$$

and

$$arg = \frac{a-2(a+b')\psi_0^2 r_0^2/r^2}{(a^2+4b'(a+b')\psi_0^2)^{1/2}}$$

where

$$\psi_0 \simeq \sin\psi_0$$

$$b' = br_0^2/R_z^2$$

Note the angle ψ_0 is very small.

These arguments were then reduced using a binomial expansion:

$$arg_0 = 1 - \frac{2(a+b')^2\psi_0^2}{a^2}$$

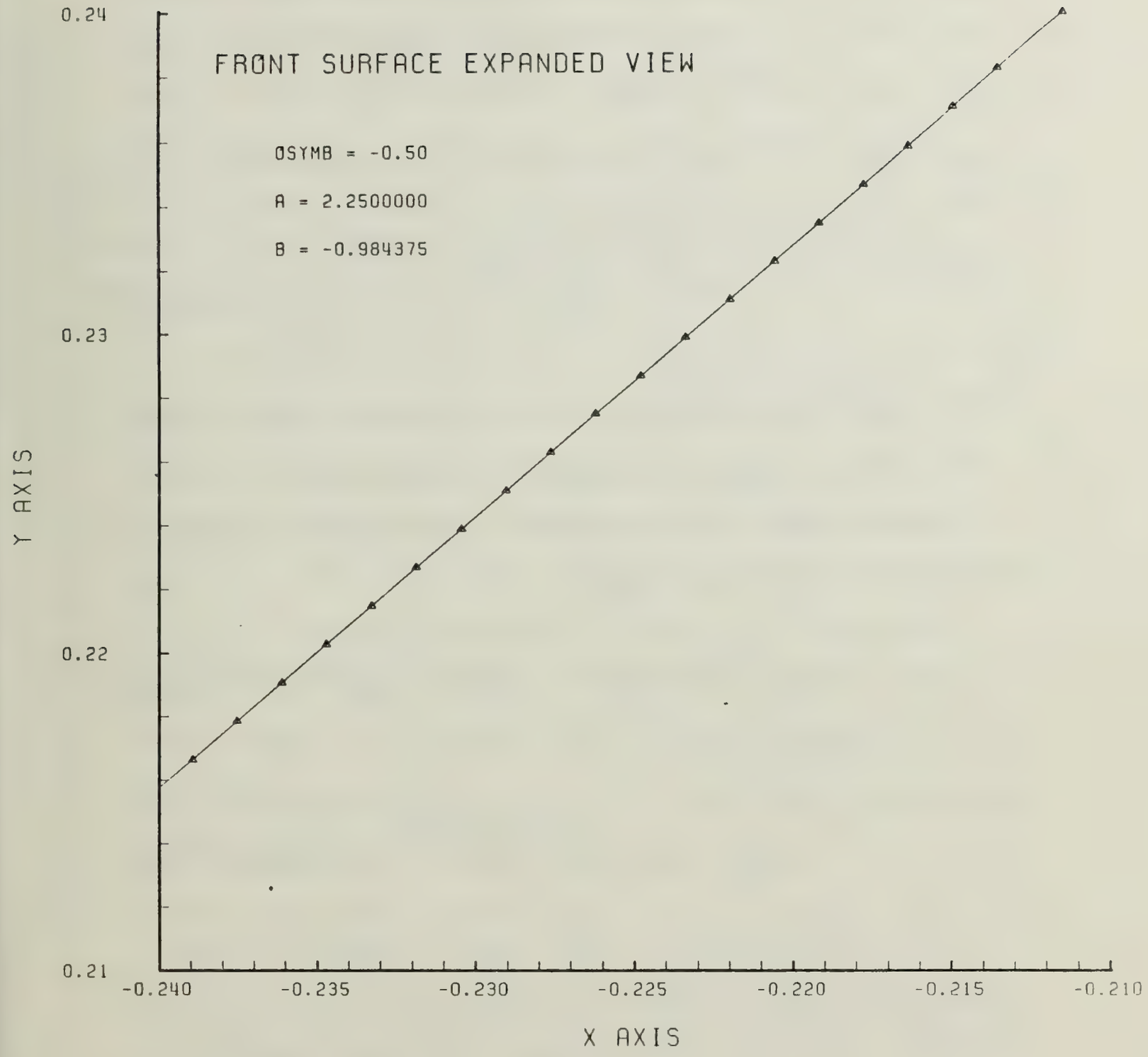

Figure 3.5 Expanded View of Front Surface - Straight Line Method.

and

$$arg = 1 - \frac{2(a+b')\psi_0^2\left(\frac{r_0^2}{r^2}\, a+b'\right)}{a^2}$$

Again the ψ_0^2 term ensures that the expansion is valid.

A function of the type arcsin(x) where x is known to be close to the value 1 can be curve fitted into the form:

$$\sin^{-1}(x) = \pi/2 - [2(1 - x)]^{1/2}$$

This equation applies to both arcsine functions as the ψ_0^2 term ensures that the argument is close to 1.

When placed in the equation the arguments result in the following final expression:

$$\theta = \theta_0 + \frac{\varepsilon\psi_0}{2}\left\{(a+b') - \left[(a+b')\left(\frac{r_0^2}{r^2}\, a+b'\right)\right]^{1/2}\right\}$$

In terms of radius:

$$r = r\,\frac{(a(a+b'))^{1/2}}{\left\{\left[\frac{a}{\varepsilon\psi_0}\,(\theta - \theta_0) - (a+b')\right]^2 - b'(a+b')\right\}^{1/2}}$$

The reduced equation was then implemented in the iterative front surface solution scheme by constantly checking it against the complete equation. When the

difference between the two became small, the reduced equation was subsequently used in solving for the front surface. The reduced equation was then turned off when the rays had progressed an equal distance on the other side of the singularity. See Figure 3.2 for clarification.

The equation worked well in resolving the front surface for the lens construction and did produce a smooth surface similar to that shown in Figure 3.5.

It did not do well when the accuracy program check was applied. One reason is that although it was easy to implement the equation for the lens construction it was not so easy to complete the forwards trace since these rays hit the front surface randomly making it difficult to know when to use the reduced equation. In practice the reduced equation could be implemented when the absolute value of the parameter ψ_0 became smaller than that used in the lens construction phase during the time that the reduced equation was used. There was then no way of telling how accurate the front surface was, in other words, there was no method to show how the rays would physically react in the region of the singularity. Thus the reduced equation method would work but it is not an entirely satisfactory solution.

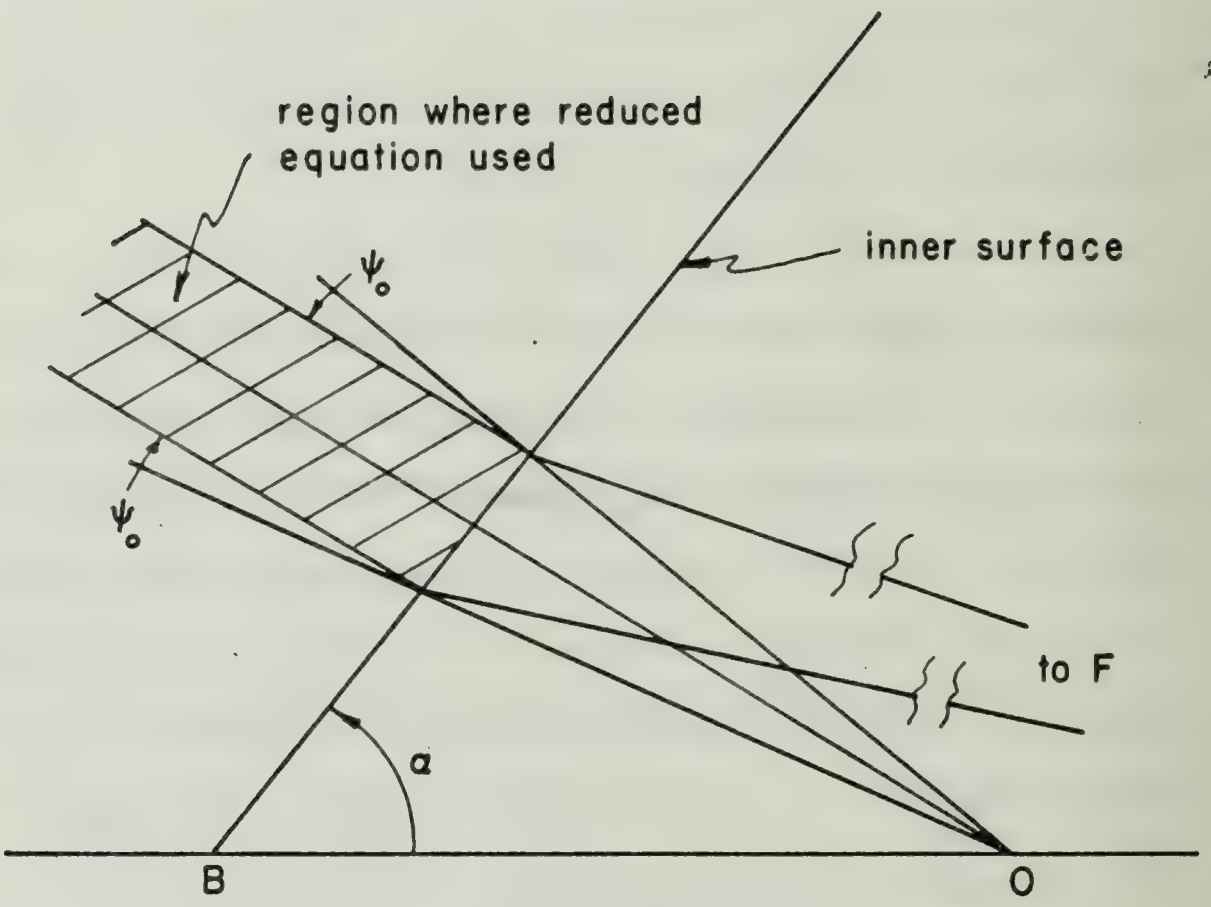

Figure 3.6 Reduced Equation Implementation.

Program CHECK was tried without using the reduced equation but disastrous results were obtained for those points in the region of the singularity.

For these reasons it was decided that another means of resolving the problem was required.

G. THIRD PROPOSED SOLUTION - MARCHAND METHOD

The third proposed solution is drawn from Marchand [Ref. 5] p. 27, in which he proposes a means of improving the ray trace formula so that nearly radial lines will be more easily described.

The method is based on redefining variables as follows:

$$\theta = eM_0 \tag{3.1}$$

where

$$M_0 = \int_{r_0}^{r} \frac{dr}{r(n^2r^2 - e^2)^{1/2}}$$

Marchand describes the ray path using the parameters α and β such that:

$$\beta = \sin\theta/\sin\psi_0$$

$$\alpha = \cos\theta - \beta\cos\psi_0 \tag{3.2}$$

This enables the coordinates of the ray to be described as:

$$x = r\left(\alpha\frac{x_0}{r_0} + \beta\frac{p_0}{n_0}\right)$$

$$y = r\left(\frac{y_0}{r_0} + \frac{q_0}{n_0}\right)$$

$$z = r\left(\frac{z_0}{r_0} + \frac{0}{n_0}\right)$$

Upon substitution of equations 2.2 and 3.2, equation 3.1 becomes:

$$= n_0 r_0 M_0 \text{ sinc}$$

The parameter can now be more accurately solved for small angles of using a Maclaurin series for the sinc function.

This method was then implemented in program GISL using an iteration scheme based on Cartesian rather than polar coordinates.

This method was not found to improve the accuracy of the ray trace in the region of near radial lines and was not adopted.

One reason for this may lie in the fact that in our solution to the integral in equation 2.1 the value e no longer appears at the front of the expression. Thus there is no advantage to redefining the variable as shown in equation 3.1.

H. FOURTH PROPOSED SOLUTION - ANGLE IN TERMS OF RADIUS

The fourth method and the one that proved successful involved the solution of the front surface slope and the

GRIN ray using the form of the ray trace equation shown in equation 2.3 rather than that of equation 2.4.

The reason for this choice is shown in Figure 3.7. This figure shows that when the rays are nearly radial a small change in angle will result in a huge change in radius. Thus the form of the ray tace equation using r( ) is inherently unstable. Previously we have used this form in order to make use of the Newton-Raphson technique. It was thought that if the iteration scheme could be reworked so that the form θ(r) of the equation could be used then better results might be obtained.

A suitable iteration method was found and is illustrated in Figures 3.8 and 3.9.

The figures show that an initial point was chosen in the same manner as before, that is by drawing a straight line from the ray refracted at the inner surface out to the front surface slope. This point provides the radius for equation 2.4 so that a new angle can be calculated. This angle is then used to find the corresponding radius out to the front surface slope. This new radius is then used in equation 2.4 to find the next angle and thus an iterative loop is formed.

Thus a simple scheme was found. It should be noted however that this method will only converge when the initial angle is greater than 90 degrees. The reason for this lies in the geometry and is shown in Figure 3.10.

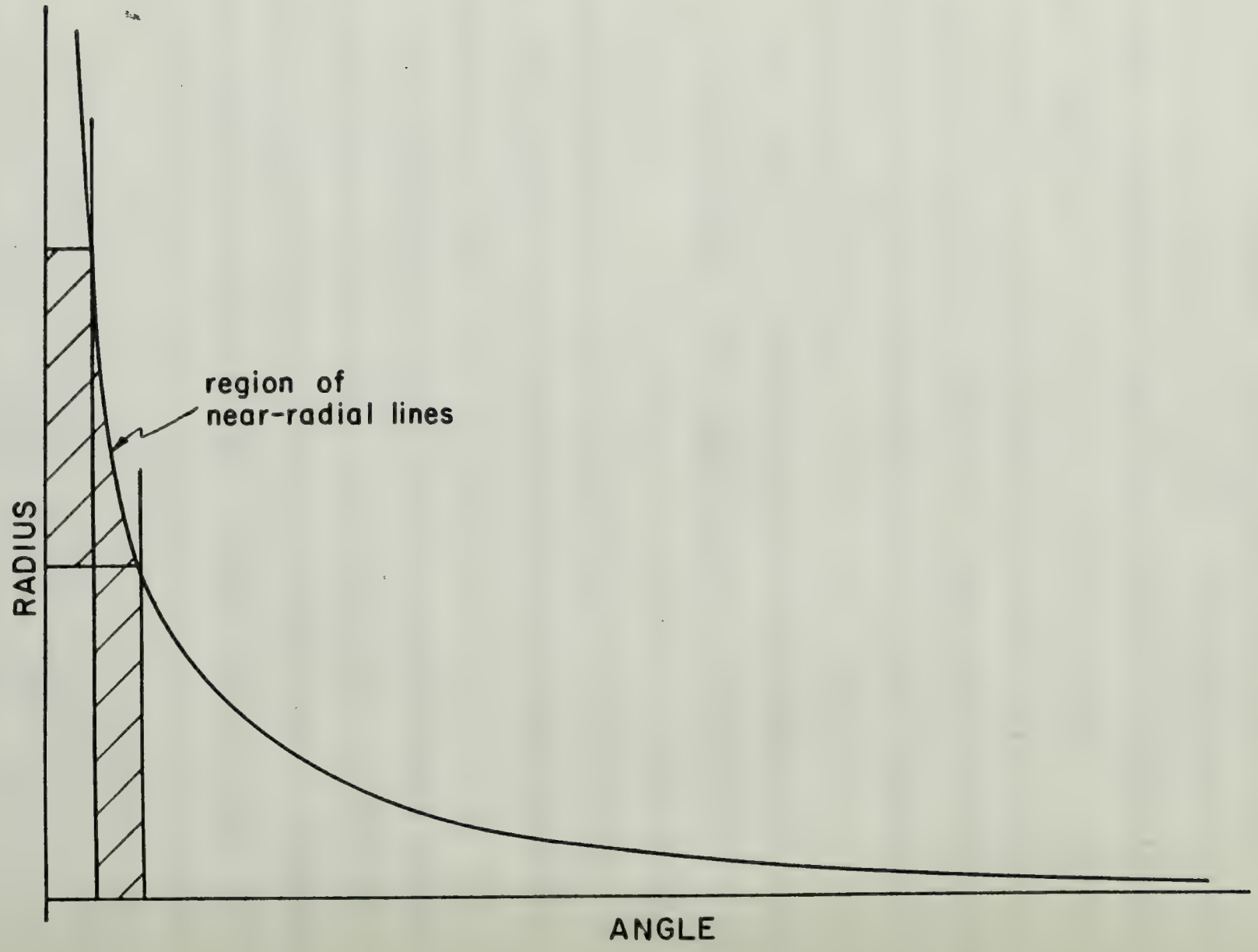

Figure 3.7 Radius and Angle Relation.

Thus the iteration will only converge when the rays are in the region of the nearly radial lines. This does not cause a difficulty since the Newton-Raphson method, modified to include ray matching, can find solutions efficiently in the region where the initial angle is less than 90 degrees.

Figure 3.10 is also drawn to show the reader an example of a forward ray trace.

This simple scheme was implemented and found to work successfully for both the lens construction and the skew ray trace; see [Ref. 8].

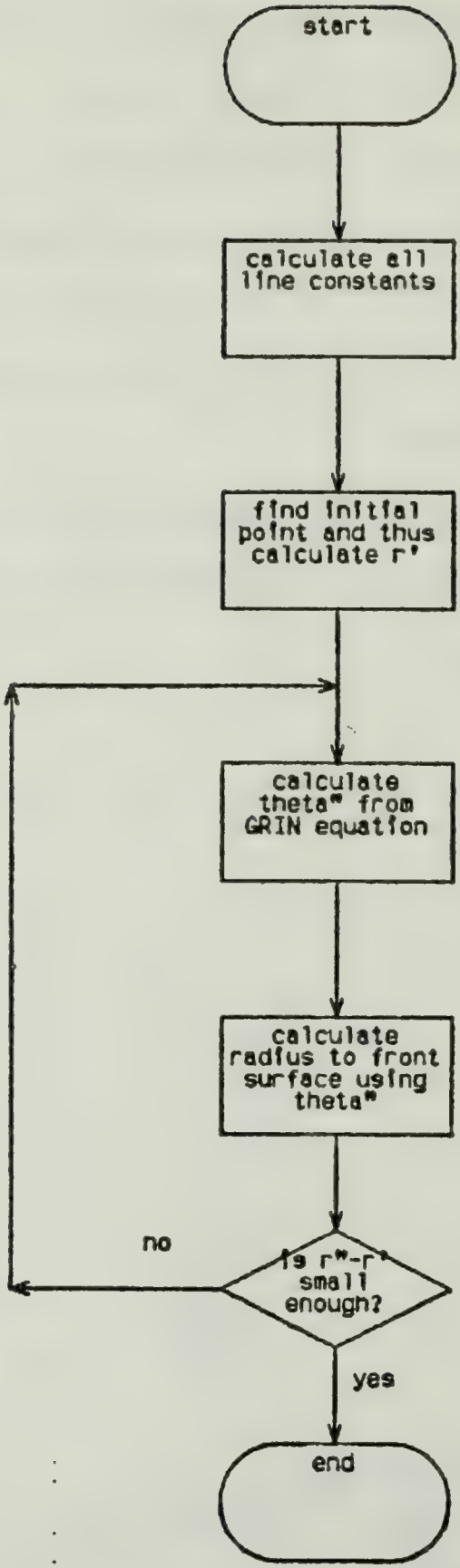

Figure 3.8 Angle in Terms of Radius Iteration Logic Flowchart.

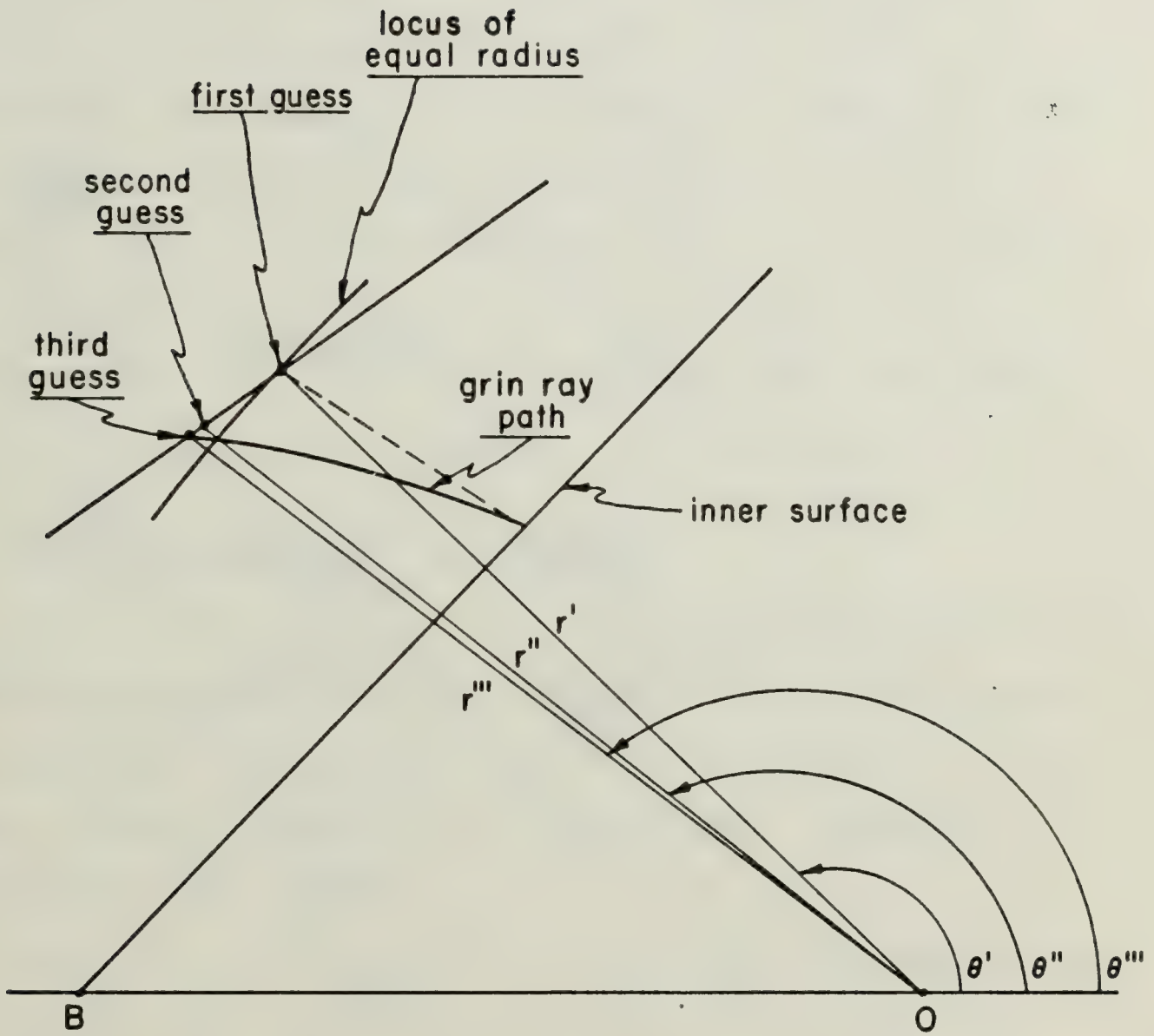

Figure 3.9 Angle in Terms of Radius Iteration Geometry.

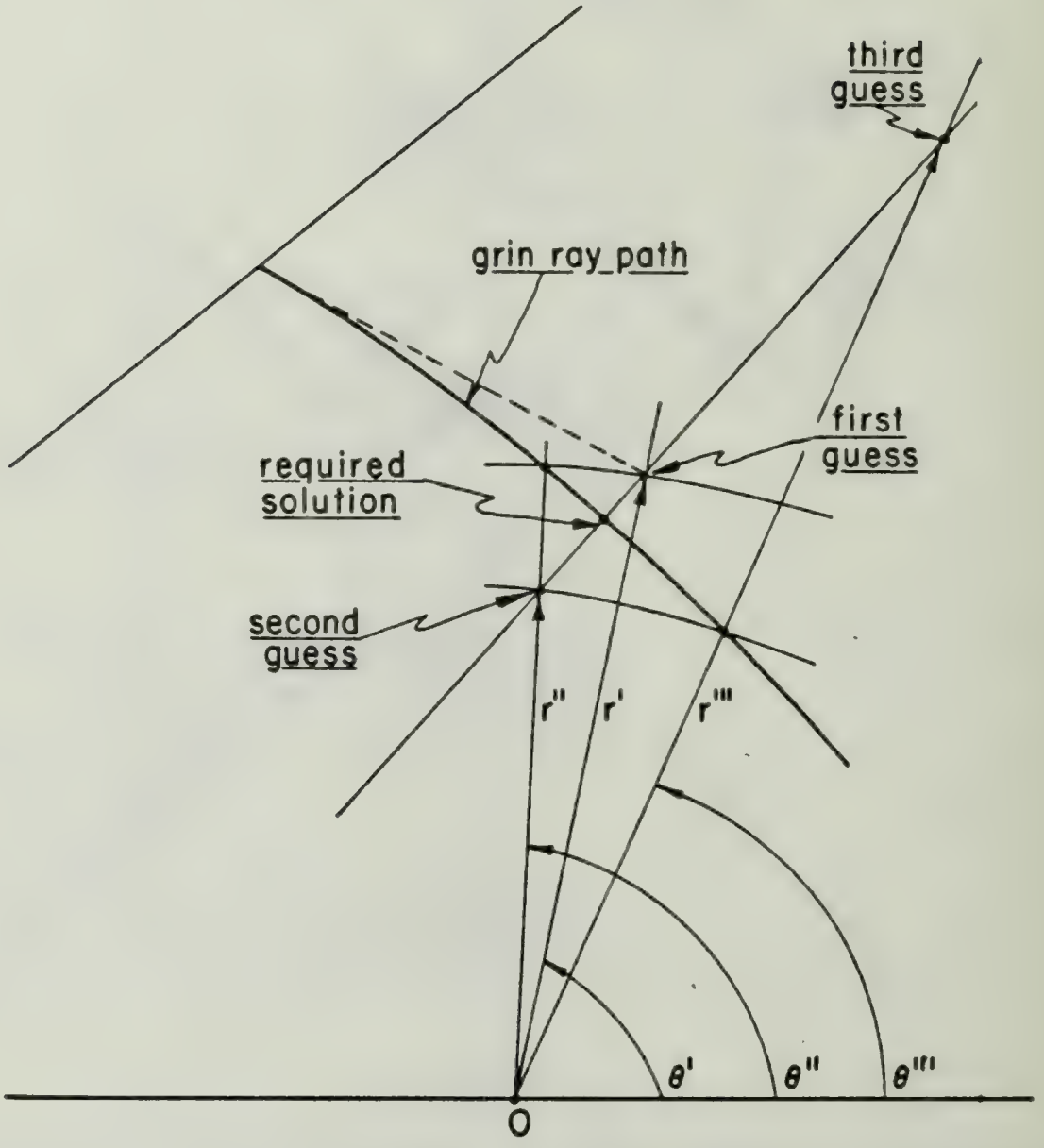

Figure 3.10 Diverging Iteration - Forward Ray Trace.

IV. GRADIENT INDEX SEEKER LENS RAY TRACE RESULTS

A. FRONT SURFACE ACCURACY

As mentioned previously, the front surface of the Gradient Index Seeker Lens was checked for accuracy by completing a ray trace from the midpoint of each set of solutions. A sample of the output from program Check is shown in Table I so that the reader will have an indication of the accuracy of the ray trace in the region of near-radial lines. Note that the most radial lines occur when the absolute value of the angle ψ_0 is smallest.

B. SPOT SIZE RESULTS

The results of program GISL showing the variation of spot size are shown in Figure 4.1. This figure shows how the spot size varies as gradient strengths and orientations of the center of symmetry are changed for a median angle of incidence of 0.3 radians.

Note that all orientations of the center of symmetry are now included. The studious reader should refer to Carr's work [Ref. 4], to compare results.

C. SELECTION OF A BEST LENS

The best lens was selected by determining those parameters that would result in a small spot size and also cause the least number of rays to be reflected or refracted

outside the missile diameter. Figure 4.1 shows that these criterion were not easily discernable and a judgemental decision was required.

The reader should note that a value of 2.25 was chosen for the parameter a. This was done since the resulting value of 1.5 for the index of refraction at the center of symmetry is common to a large number of possible lens materials. Also of note is the fact that the focal point was chosen to be 2.0 units along the x-axis from the inner surface cone. This was done since this value resulted in smaller spot sizes generally.

With these criterion in mind a value of -0.2 for OB and a 5% negative gradient were chosen as parameters for the best lens design.

A more detailed study of the lens parameters could be accomplished with use of an optimization program in which a large number of parameters could be varied in order to find the best lens. This would achieve a more satisfactory result than the one shown.

TABLE I

FRONT SURFACE ACCURACY

Line Number	Angle θ_0 (Radians)	Error in y-coordinate	number of iterations
895	0.0411	0.0008	4
896	0.0390	0.0007	4
897	0.0369	0.0008	4
898	0.0349	0.0009	4
899	0.0328	0.0007	4
900	0.0308	0.0009	4
901	0.0288	0.0008	4
902	0.0268	0.0008	4
903	0.0247	0.0009	4
904	0.0227	0.0009	3
905	0.0207	0.0008	3
906	0.0187	0.0009	3
907	0.0167	0.0008	3
908	0.0147	0.0008	3
909	0.0127	0.0010	3
910	0.0107	0.0009	3
911	0.0087	0.0008	3
912	0.0067	0.0010	3
913	0.0048	0.0009	3
914	0.0028	0.0010	2
915	0.0008	0.0010	2
916	-0.0010	0.0010	2
917	-0.0030	0.0010	3
918	-0.0050	0.0010	3
919	-0.0069	0.0009	3
920	-0.0089	0.0011	3
921	-0.0108	0.0010	3
922	-0.0127	0.0010	3
923	-0.0147	0.0009	3
924	-0.0166	0.0011	3
925	-0.0185	0.0010	3
926	-0.0204	0.0011	4
927	-0.0223	0.0011	4
928	-0.0243	0.0010	4
929	-0.0261	0.0012	4
930	-0.0281	0.0011	4
931	-0.0300	0.0012	4
932	-0.0318	0.0012	4
933	-0.0337	0.0012	4
934	-0.0356	0.0011	4
935	-0.0375	0.0013	4

$\frac{\Delta N_{\%}}{N}$	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0	-.1	-.2	-.3	-.4	-.5	-.6	-.7	-.8	-.9	-.10
+40	.234 8,107	.225 12,105	.206 16,101	.212 17,100	.217 14,100	.219 14,100	.221 14,100	.223 12,98	.230 12,100	.233 12,101	.243 8,82	.216 14,97	.222 10,100	.210 18,91	.295 19,98	.202 24,92	.283 22,95	.295 24,83	.274 34,73	.309 26,77	.408 34,69
+35	.220 12,103	.217 14,103	.205 16,101	.212 10,100	.216 15,100	.219 15,100	.221 14,100	.224 12,98	.226 13,98	.233 12,100	.243 8,86	.217 14,97	.224 10,100	.212 18,92	.299 19,95	.250 26,95	.288 22,93	.308 28,81	.284 34,73	.324 30,71	.422 36,67
+30	.214 14,101	.217 14,101	.219 14,103	.211 16,101	.216 14,101	.222 14,101	.222 15,98	.224 13,98	.240 11,102	.233 12,101	.234 10,77	.218 10,99	.220 10,98	.215 20,94	.219 23,91	.218 19,96	.294 20,93	.230 30,83	.316 32,73	.342 32,68	.436 36,67
+25	.220 14,101	.221 14,101	.222 14,101	.224 14,103	.215 14,101	.219 14,99	.222 14,99	.230 14,101	.238 10,103	.233 12,101	.230 6,82	.220 10,99	.217 10,95	.218 18,94	.221 21,92	.220 23,93	.333 22,91	.328 30,81	.345 30,79	.379 36,72	.455 38,67
+20	.241 12,105	.230 14,103	.226 14,101	.227 14,101	.229 12,101	.234 12,103	.235 12,103	.236 12,103	.236 12,103	.249 10,105	.231 6,78	.217 12,99	.221 10,93	.218 18,92	.234 18,98	.253 23,89	.231 29,83	.310 33,74	.268 33,72	.432 37,72	.377 42,65
+15	.245 12,105	.234 14,103	.234 14,103	.235 14,103	.235 12,103	.235 12,103	.235 12,103	.235 12,103	.245 10,105	.247 10,105	.235 4,79	.220 12,98	.215 12,93	.227 16,93	.235 16,94	.249 21,94	.242 27,85	.330 33,76	.282 35,71	.466 43,65	.365 47,61
+10	.250 12,105	.250 12,005	.250 12,005	.249 12,105	.249 12,105	.239 12,103	.248 10,105	.247 10,105	.246 10,105	.246 10,105	.233 10,81	.239 8,100	.233 10,99	.239 10,97	.235 12,91	.253 20,84	.264 27,86	.279 37,78	.301 39,74	.350 47,66	.425 47,64
+ 5	.256 12,105	.255 12,105	.255 12,105	.254 10,105	.254 12,105	.253 12,105	.251 12,105	.250 12,105	.247 12,105	.244 12,105	.248 8,91	.238 10,10	.234 10,99	.233 12,97	.272 8,95	.268 26,89	.300 30,83	.309 35,73	.378 39,65	.395 47,63	.499 45,59
0	.262 12,105	.262 12,105	.261 12,105	.261 12,105	.260 12,105	.259 12,105	.257 12,105	.255 12,105	.251 12,105	.241 12,105	.246 12,104	.237 9,100	.290 9,100	.240 11,98	.247 12,91	.306 20,83	.340 22,79	.333 30,73	.426 39,64	.425 39,64	.588 37,60
- 5	.240 14,107	.239 14,107	.238 14,107	.237 14,107	.237 14,107	.236 14,107	.236 14,107	.265 14,107	.256 14,105	.238 14,111	.228 8,83	.238 9,99	.228 11,98	.230 13,94	.312 15,88	.250 21,84	.264 29,76	.385 39,71	.356 41,71	.399 47,59	.505 47,51
-10	.240 15,109	.239 16,109	.237 16,104	.236 16,109	.240 14,111	.239 14,111	.238 14,111	.238 14,111	.238 14,111	.247 14,111	.212 10,83	.245 8,100	.229 8,98	.226 8,92	.243 12,87	.232 25,78	.266 20,73	.401 36,53	.461 46,53	.474 41,45	.579 50,49
-15	.263 17,111	.262 17,111	.261 16,111	.260 16,111	.256 16,111	.256 16,111	.256 10,111	.253 16,111	.250 14,111	.246 14,111	.243 10,87	.256 6,102	.231 8,98	.270 6,97	.328 17,84	.242 10,70	.322 8,65	.287 22,55	.342 20,56	.389 36,42	.462 38,40
-20	.297 18,110	.298 18,110	.296 19,109	.292 18,109	.289 18,109	.289 18,109	.289 18,109	.285 18,109	.280 18,109	.287 18,111	.267 10,83	.271 4,103	.236 8,103	.237 6,86	.338 6,88	.324 8,80	.371 0,65	.390 10,52	.462 14,46	.498 16,40	.515 13,57
-25	.294 16,111	.293 16,111	.291 16,111	.291 15,112	.290 12,114	.294 12,114	.294 12,114	.290 12,114	.280 14,112	.298 14,114	.266 10,92	.277 7,103	.289 6,96	.274 6,92	.334 2,89	.351 2,75	.424 0,67	.418 6,58	.463 4,45	.502 10,39	.556 14,34

OSYMB

Figure 4.1 Spot Size Results - ALFAP = 0.3 Radians. Each Box Contains Spot Size, Number of Rays Outside Missile Diameter, Number of Rays on Image Plane. Total Number of Incident Rays is 135. Number of Rays Reflected Inside Lens Not Shown.

V. MIRROR AND DETECTOR SYSTEM

A. DESCRIPTION OF SYSTEM

A scanning mirror and a detector were added to the GRIN best lens in order to determine if precise definition of off-axis targets would be possible. The design of these two new elements is shown in Figure 5.1.

The mirror is made to 'nod' at a high rate and thus scans the spot across the detector. The detector consists of two sections, as shown in Figure 5.1. Thus, when the spot is scanned across the two plates a signal from each detector will be generated and will result in the form shown in Figure 5.2. This arrangement will allow an accurate measurement of the mirror angle for which the spot crosses the axis. The electronics of the detector signal processor could then be arranged to relate the mirror angle to the angle of incident radiation so that the precise line-of-sight angle of an off-axis target will be known.

Note that this is a simple detector system that will only resolve angles in one plane. A more sophisticated system along the same lines could be designed with a four-element detector and a more intricate method of scanning the spot across the detector. It was decided to keep the mirror and detector system as simple as possible since the

aim of this work is only to show that a precise angle measuring device is feasible.

B. LENS TO DETECTOR RAY TRACE

This section describes the mathematics of the ray trace from the lens to the mirror and then to the detector. A listing of program DETPLOT which accomplishes this task is included as Appendix C.

The ray trace consists of the following parts:

a. ray direction exiting lens

b. point of intersection on mirror

c. ray direction exiting mirror

d. detector ray count

1. Ray Direction Exiting Lens

The ray direction exiting the lens is derived from the direction cosines of the ray as calculated by program GISL. This data is used as an input to program DETPLOT along with the coordinates of the point at which the ray left the lens.

2. Point of Intersection on Mirror

The point of intersection of the ray and mirror may be described by the following expressions:

$$x_m = D_{2m}K + x_0 \qquad (5.1)$$

$$y_m = D_{2m}L + y_0 \qquad (5.2)$$

$$z_m = D_{2m}M + z_0 \qquad (5.3)$$

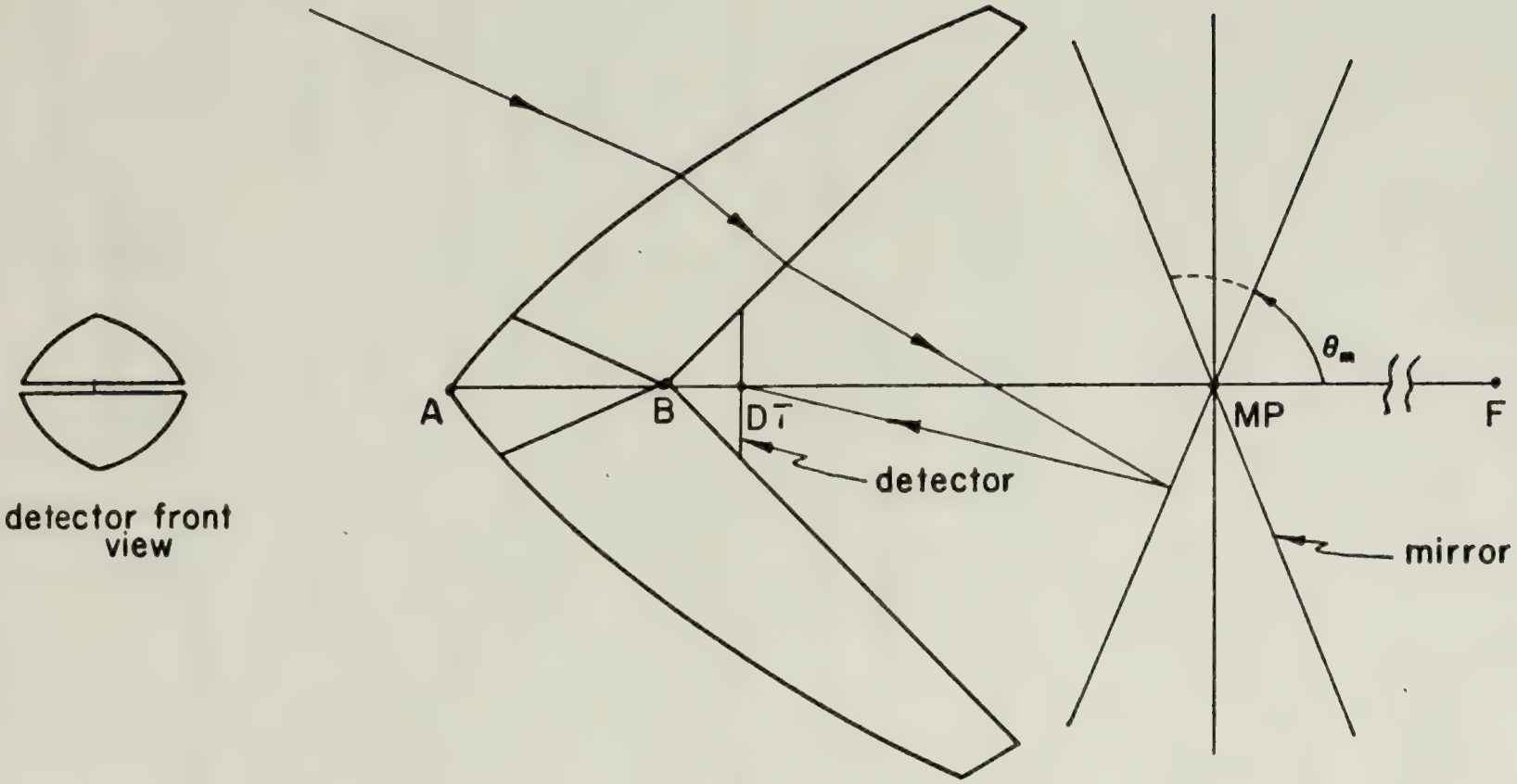

Figure 5.1 Lens, Mirror and Detector System.

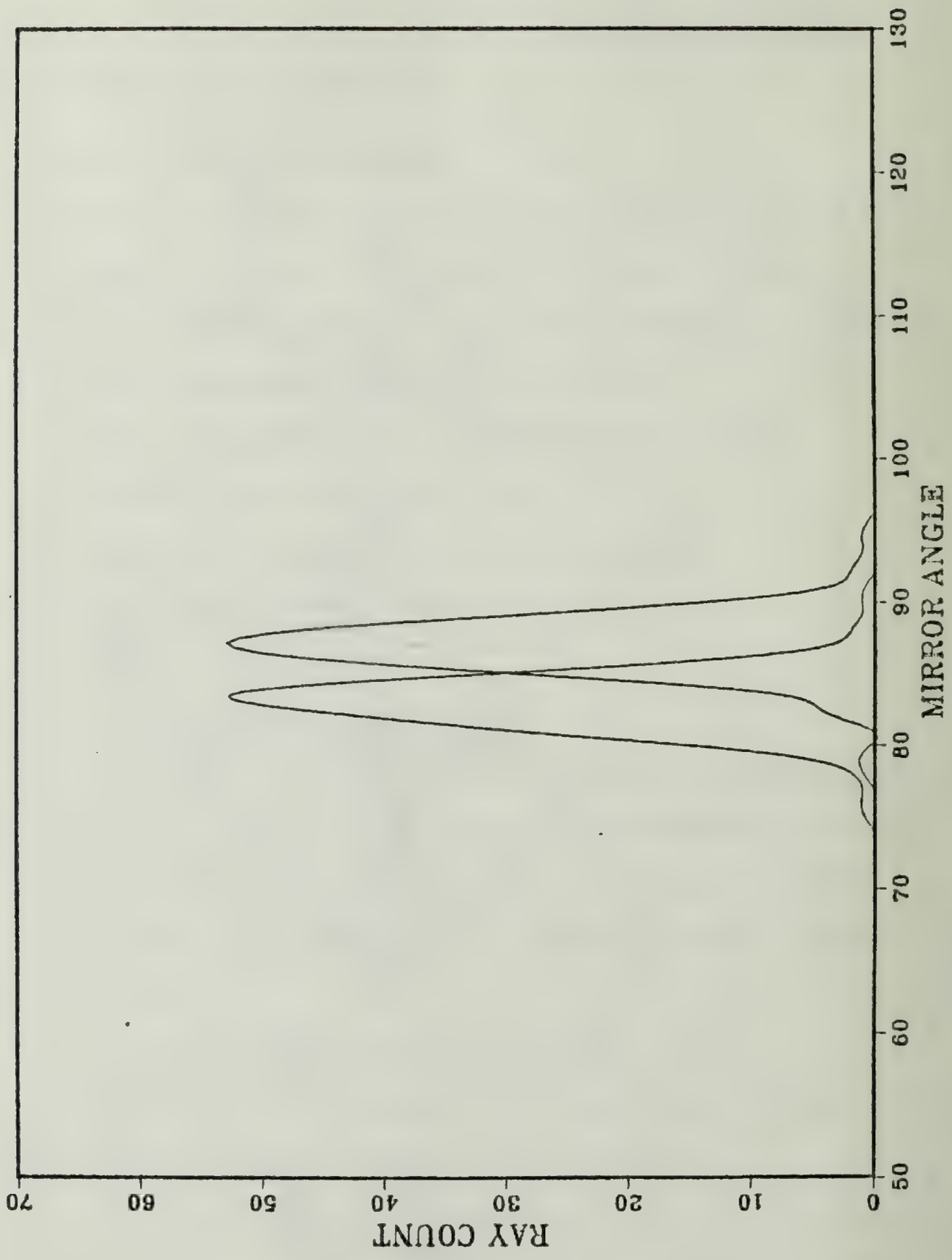

Figure 5.2 Detector Signal.

Since the surface of the mirror is tilted along the z-axis it can be described by the equations:

$$y_m = \tan\theta_M(x_m - MP) \qquad (5.4)$$

$$y_m^2 \cos^2 \vartheta_m + z_m^2 \leqslant R \qquad (5.5)$$

Where equation 5.4 describes a tilted plane and equation 5.5 limits the plane to the mirror diameter.

Neglecting equation 5.5 for a moment, equations 5.1, 5.2 and 5.4 may be combined to solve for $D_{\ell m}$. The result is:

$$D_{\ell m} = \frac{\tan\ \theta_m(x_0 - MP) - y_0}{L - K \tan\ \theta_m}$$

Now that $D_{\ell m}$ is known, equations 5.1, 5.2 and 5.3 may be used to find the coordinates of the mirror intersection point. These are then checked with equation 5.5 to make sure they are not outside the mirror diameter.

3. Ray Direction Exiting Mirror

The solution of the ray direction leaving the mirror is best understood vectorially. See Figure 5.3.

Reflection off a mirror is governed by two factors. Firstly, the angles of incidence and reflection must equal. Using vectors this is expressed as:

$$\vec{n} \cdot \vec{r}_i = - \vec{n} \cdot \vec{r}_r$$

In component form:

$$n_x r_{xi} + n_y r_{yi} = - n_x r_{xr} - n_y r_{yr} \qquad (5.6)$$

Secondly, the ray is reflected in the same plane as the incident ray. This may be expressed vectorially by using the cross product to generate the vector $\vec{n}_p$ as shown in Figure 5.3. Thus:

$$\vec{n}_p = \vec{r}_i x \vec{n} = \vec{r}_r x \vec{n}$$

In component form:

$$\vec{e}_x(-n_y r_{zi}) + \vec{e}_y(-n_x r_{zi}) + \vec{e}_z(n_y r_{xi} - n_x r_{yi}) =$$

$$\vec{e}_x(-n_y r_{zr}) + \vec{e}_y(-n_x r_{zr}) + \vec{e}_z(n_y r_{xr} - n_x r_{yr})$$

Three equations result:

$$-n_y r_{zi} = -n_y r_{zr}$$

$$-n_x r_{zi} = -n_x r_{zr}$$

$$n_y r_{xi} - n_x r_{yi} = n_y r_{xr} - n_x r_{yr} \qquad (5.7)$$

The first two results are not unexpected as the mirror should not effect the z-component of the ray.

Equations 5.6 and 5.7 may now be combined to find the x- and y-components of the ray. When this is done the two following expressions result:

$$r_{xr} = \frac{(n_y^2 - n_x^2)}{(n_x^2 + n_y^2)} r_{xi} - \frac{2n_x n_y}{(n_x^2 + n_y^2)} r_{yi}$$

$$r_{yr} = \frac{(n_x^2 - n_y^2)}{(n_x^2 + n_y^2)} r_{yi} - \frac{2n_x n_y}{(n_x^2 + n_y^2)} r_{xi}$$

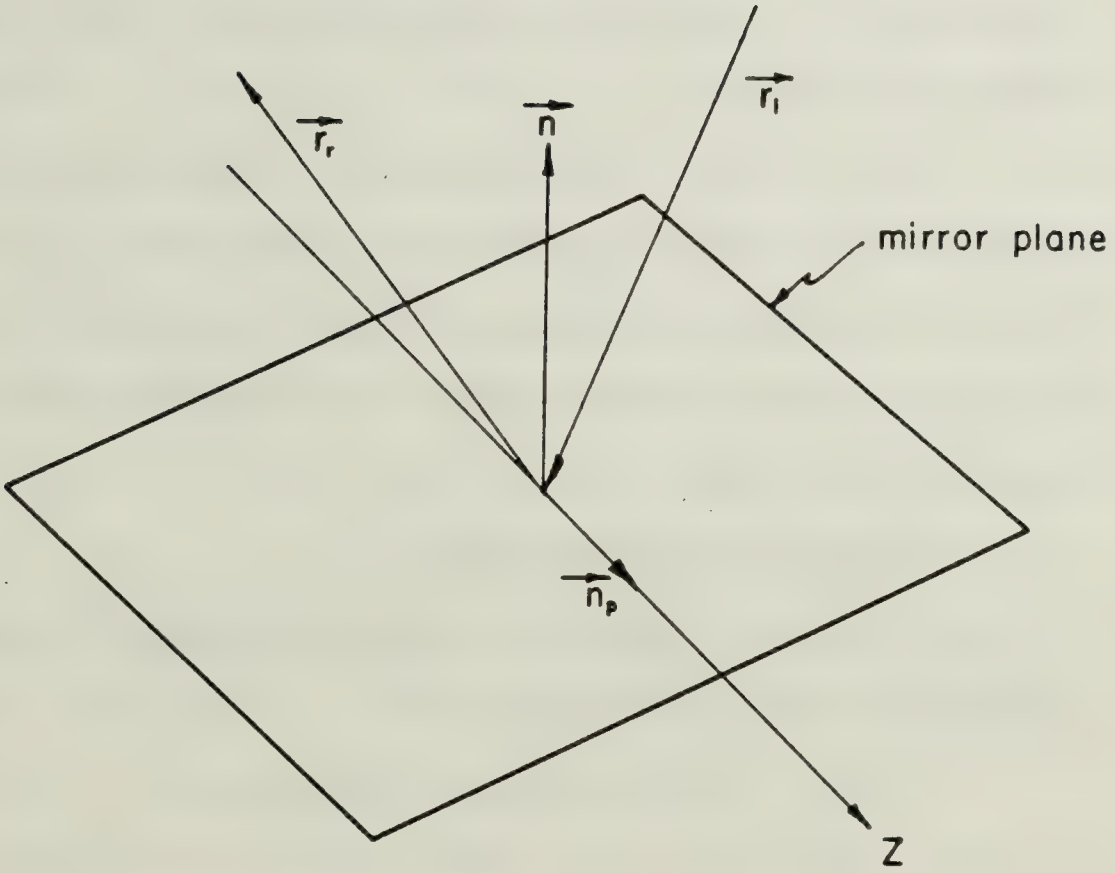

Figure 5.3 Reflection of Ray on Mirror.

Since the vector $\vec{n}$ is entirely in the x-y plane, its components may be related to mirror angle as follows:

$$n_x = -\sin\theta_m$$

$$n_y = \cos\theta_m$$

When the above equations, along with well known trigonometric identities are used the following expressions will be found.

$$r_{xr} = \cos(2\theta_m)r_{xi} + \sin(2\theta_m)r_{yi}$$

$$r_{yr} = -\cos(2\theta_m)r_{yi} + \sin(2\theta_m)r_{xi}$$

Thus these equations represent the change in direction for the exiting ray and form two of its direction cosines, the third being r_{zr}.

4. Ray Count on Detector

The coordinates of the ray at the detector are described by the expressions:

$$x_d = D_{nd}\ K' + x_m$$

$$y_d = D_{nd}\ L' + y_m$$

$$z_d = D_{nd}\ M' + z_m$$

Since x_d is used to define the position of the detector, it is known. Thus the value D_{nd} may be solved and the corresponding y and z-components may be found.

The ray is then checked to make sure it is within the detector diameter. If it is the y-coordinate is checked to find out if it is negative or positive, and thus

whether the ray has struck the upper or lower half of the detector. The ray count is then incremented accordingly.

Program DETPLOT traces all rays for each increment of mirror angle and thus generates plots such as that shown in Figure 5.2.

C. MIRROR AND DETECTOR SYSTEM RESULTS

Appendix D shows the results of the detector signals as a function of mirror angle at angles of incidence from 0 to 40 degrees. For these the mirror pivot point was placed at 1.4 units along the x-axis from the inner surface cone since this was the smallest distance that would still allow full movement of the scanning mirror. The focal length was adjusted to 2.8 units for the lens design calculations so that the spot would be imaged as small as possible on the detector. Figure 5.4 summarizes the results shown in Appendix D. As can be seen, the smaller angles of incidence (those in the range 0 to 16 degrees) result in a linear relation between angle of incidence and mirror tilt angle, corresponding to an on-axis spot. The larger values of the angle of incidence resulted in a more random relation between these two quantities.

Appendix D also shows that the error in determining mirror angle increases as the angle of incidence increases. This is due to the fact that each successive plot is less distinct and thus a precise measure of the angle becomes

more difficult. Although this effect shows a deterioration in the lens system performance, it is not harmful since, as the missile corrects itself, the measurement of angle will become more accurate.

For these reasons this simple system is considered to be a viable means of target detection and shows that the GISL can be a successful missile seeker lens.

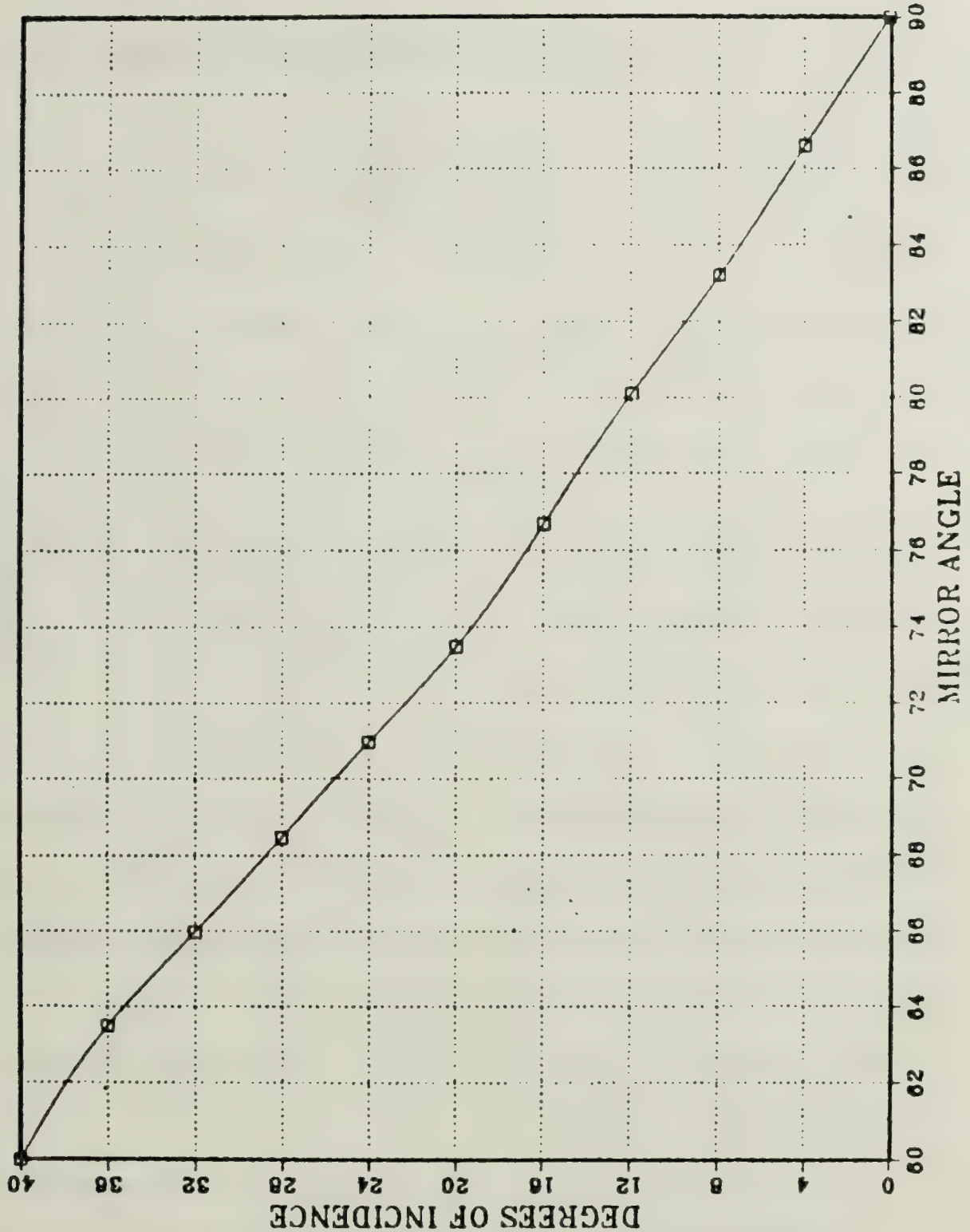

Figure 5.4 Mirror Tilt for On-Axis Spot and Angle of Incidence.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Two objectives were accomplished in the course of this study. Firstly, the Gradient Index Seeker Lens design, as initiated by Carr [Ref. 4], was completed in that all orientations of the center of symmetry with respect to the lens were made possible to describe mathematically. The results obtained for the spot size on the image plane showed trends that indicated the parameters that would produce a 'best' lens, but precise definition of these values was not possible.

Secondly, it was shown that a scanning mirror and detector system applied to the lens could be described using ray-trace techniques and also that such a system could be used to provide precise angle measurements of off-axis targets at angles less than 16 degrees. For larger values of angle, error in definition became increasingly greater.

Specifically, then, the following conclusions are made:

a) The Gradient Index Seeker Lens can be described using ray-trace techniques and can be shown to provide adequate imaging of off-axis objects.

b) A scanning mirror and detector system can be used to provide precise angle measurements.

B. RECOMMENDATIONS

The following recommendations for further study are made:

a) An optimization program should be applied to program GISL to study the variations of all important parameters.

b) A detector system capable of detection in all directions should be designed.

c) The manner in which signal processing of the detector signals can be carried out should be studied.

d) A study of Gradient Index material and the methods by which a lens could be manufactured should be undertaken.

```
C     ***********************************************************GISOC010
C     *                                                         *GISOC020
C     *                  GRADIENT INDEX SEEKER LENS             *GISOC030
C     *                  /      /      /      /                 *GISOC040
C     *                                                         *GISOC050
C     ***********************************************************GISOC060
C     *                                                         *GISOC070
C     *        THIS PROGRAM DESIGNS THE GRADIENT INDEX SEEKER LENS *GISOC080
C     *    IN TWO STAGES.   FIRST THE FRONT SURFACE IS CONSTRUCTED *GISOC090
C     *    BY TRACING RAYS FROM THE FOCAL POINT BACK THROUGH THE *GISOC100
C     *    LENS DEFINING THE FRONT SURFACE SO THAT THE RAYS EMERGE *GISOC110
C     *    PARALLEL.   SECOND A SKEW RAY TRACE IS COMPLETED IN A *GISOC120
C     *    FORWARD DIRECTION FOR ANY ANGLE OF INCIDENCE.   THE SIZE *GISOC130
C     *    OF THE RESULTING SPOT PATTERN IS RECORDED SO THAT     *GISOC140
C     *    PERFORMANCE MEASUREMENTS MAY BE MADE.                 *GISOC150
C     *        THIS PROGRAM WAS INITIALLY WRITTEN BY H. CARR AND *GISOC160
C     *    THEN MODIFIED BY D. DAVIDSON                          *GISOC170
C     *                                                         *GISOC180
C     ***********************************************************GISOC190
C                                                                 GISOC200
C   SPECIFICATION STATEMENTS                                      GISOC210
C                                                                 GISOC220
      INTEGER I,J,K,L,G,F,M,N,P,FLAG(1000),Q,RAYS,S,DRAYS,ELLNUM,  GISOC230
     1NUMBR,CCORD,ENUM,NUMB,IANGLE,THICK,GG,HF,COUNT,SHAPE,GGT,   GISOC240
     1CCOR(800),NUMBT,ELLIPS,SQURAY,TRIP,FILTER,CCORT,            GISOC250
     1FACE,FLLG,COUNT2                                           GISOC260
      REAL DYDXN(1010),DYDXT(1010),I1P(1010),GAMMA,AB,ONP,       GISOC270
     1T,ALFAF,GRIC,RAYY(1000),RAYZ(1000),CENI,TORBI,BOTI,TOPI,   GISOC280
     1RADIUS,CEN,TOP,BOT,TORB,DELTAX,DELTAY,SF,ST,RATIO,DYDXNP,LK,LL GISOC290
      REAL NPI,NPJ,NPK,AO,BO,CO,DO,EC,PAR1,PAR2,NINCTY(1000),    GISOC300
     1ELL,ELY(7,1000),ELZ(7,1000),ELZZ,LM,ZOYC,                  GISOC310
     1PAR3,XDIAPT(1000),YDIAPT(1000),SUM1,                       GISOC320
     1SUM2,SUM3,SIGMAZ(10,800),SIGMAY(10,800),RMSRAD(10,800)     GISOC330
      REAL RCC(500),SUM4(500),FRACTN(500),SDRAD(1000),YCENTR(20,800), GISOC340
     1THETAE,EPSLON,PSI,X1P,Y1P,X1HP,Y1HP,ERRCR,XF,YP,ZP,RH,D2T,ABS,X1PPGISOC350
      REAL Y1PP,ZETAP,BASE,BAS1,ROOO,N2X1K,N2X2K,                 GISOC360
     1PRCNTE,PRCNTT,PRCNTC,SURFL,TSURFL,SLINCR,ARKCOS,CHEKRO,CALIBR, GISOC370
     1SIGN,DIFFOR,XMAVE,XMAVET                                   GISOC380
      DOUBLE PRECISION A,B,U,F,R,ALPHA,BETA,PI,PI2,              GISOC390
     1X1(1010),Y1(1010),I1(1010),I2(1010),OSYMB,X2(1010),Y2(1010),X1H, GISOC400
     1Y1H,I2P(1010),UP(1010),UDP(1010),N1,N2,N20,N3,E,BF,RO,RZERO, GISOC410
     1DLUDP,ARG,RSINO,C,DIFF,PSIO,PSIP,ZETA,EXIT,THETAO,THETP,THETAH, GISOC420
     1RAD,RADE,RDG,RDGG,RDF,RDT,EONR,THETAP,TCRIT,THETAN,THETA,RSIN, GISOC430
     1BE,SL,RF,RPF,CHECK1,CHECK2,RADH,RDCTS,RP,THTAP,BEFF,SQRE,N2X21, GISOC440
     1XX,ART                                                     GISOC450
      DOUBLE PRECISION XC,YO,ZO,RO,PHI,PHIP,CK,CL,CKP,CLP,CMP,CKKP, GISOC460
     1CLLP,CMMP,ROX,ROY,ROZ,NPO,NPOX,NPOY,NPOZ,XX,CCK,CCL,CCM,THETA1, GISOC470
     1THETA2,NUM,NUM1,NUM2,NUM3,NUM4,NUM5,NUM6,NUM7,NUM8,OSYMPH, GISOC480
```

```
     1CPHX,OFFY,CPHZ,XINT,XINTD,BETAO,ALPHAO,XI,YI,ZI,A1,B1,C1,DP,      GISOC490
     1LKP,LLP,LMP,KKLLMM,THETAI,PHII,PHIIP,TIRA,CKPP,CLPP,CMPP         GISOC500
     1D1,D2,D3,XIH,YIH,ZIH,OPL(1000),YIM(1000),ZIM(1000),DIM(1000)     GISOC510
C                                                                      GISOC520
C    MAJOR VARIABLE DEFINITIONS                                        GISOC530
C                                                                      GISOC540
C      A           : STRENGTH OF INDEX CONSTANT                        GISOC550
C      B           : STRENGTH OF INDEX GRADIENT                        GISOC560
C      F           : FOCAL LENGTH                                      GISOC570
C      ALPHA       : INSIDE SURFACE CONE HALF-ANGLE                    GISOC580
C      U           : INCIDENT RAY OFFSET ANGLE                         GISOC590
C      I           : NUMBER OF ITERATIONS IN LENS CONSTRUCTION         GISOC600
C      R           : MAXIMUM INSIDE SURFACE RADIUS FROM X-AXIS         GISOC610
C                    NOTE THAT ALL DIMENSIONS REFERENCED TO THIS VALUE GISOC620
C      T           : THICKNESS OF LENS AT EDGE                         GISOC630
C      N1,N3       : INDICES OF REFRACTION OUTSIDE LENS                GISOC640
C      CSYMB       : DISTANCE FROM CENTRE OF SYMMETRY TO INSIDE SURFACE GISOC650
C                    ALONG X-AXIS. POSITIVE IF O TO LEFT OF B.         GISOC660
C      ALFAP       : ANGLE OF INCIDENT RADIATION FOR SKEW RAY TRACE    GISOC670
C      GRID        : GRID INCREMENT, DEFINES NUMBER OF SKEW RAYS       GISOC680
C      ENUM        : NUMBER OF ELLIPSES. MUST BE INTEGER AND ODD       GISOC690
C      ELL         : Z-COORDINATE INCREMENT FOR ELLIPSE PLOTS          GISOC700
C                                                                      GISOC710
C                                                                      GISOC720
C      OUTPUT OPTIONS:                                                 GISOC730
C                                                                      GISOC740
C      TO HAVE LENS SHAPE DATA PRINTED SET "SHAPE" TO 1. OTHERWISE     GISOC750
C      SET "SHAPE"=0.                                                  GISOC760
       SHAPE=C                                                         GISOC770
C                                                                      GISOC780
C      TO HAVE SKEW RAY AND MIRROR IMAGE SKEW RAY DATA PRINTED SET     GISOC790
C      "SQURAY" TO 1. ELSE SET "SQURAY" TO 0. (INTEGER)                GISOC800
       SQURAY=C                                                        GISOC810
C                                                                      GISOC820
C      TO HAVE ELLIPSE SHAPE DATA PRINTED SET "ELLIPS" TO 1. OTHERWISE GISOC830
C      SET "ELLIPS"=0.                                                 GISOC840
       ELLIPS=C                                                        GISOC850
C                                                                      GISOC860
C    SET CONSTANTS                                                     GISOC870
C                                                                      GISOC880
       F=4.0                                                           GISOC890
       ALPHA=C.7853982                                                 GISOC900
       U=0.000000                                                      GISOC910
       I=1008                                                          GISOC920
       R=1.0                                                           GISOC930
       T=0.05                                                          GISOC940
       N1=1.0                                                          GISOC950
       N3=1.0                                                          GISOC960
```

```
      A=2.25                                                                    GIS00970
      B=-0.984375                                                               GIS00980
      OSYMB=-0.50                                                               GIS00990
      PI=3.14159265358979 3                                                     GIS01000
      PI2=PI/2.0                                                                GIS01010
      ALFAP=0.0000                                                              GIS01020
      GRID=0.150                                                                GIS01030
      ENUM=5                                                                    GIS01040
      ELL=GRID/7.5                                                              GIS01050
C                                                                               GIS01060
CC ***************************************************************************GIS01070
CCC                                                                             GIS01080
CCC     SECTION 1.  FRONT SURFACE CONSTRUCTION                                  GIS01090
CCC                                                                             GIS01100
C ***************************************************************************GIS01110
C                                                                               GIS01120
      BETA=DATAN(F/(F-R*DCOTAN(ALPHA)))                                         GIS01130
      DLUDP=BETA/I                                                              GIS01140
      BF=F/R                                                                    GIS01150
      WRITE(4,125) BF,ALPHA,ALFAP                                               GIS01160
 125  FORMAT(' ',3F10.7)                                                        GIS01170
      WRITE(6,1900)                                                             GIS01180
C     IF(SHAPE .EQ. 1) WRITE(6,100) ALPHA,BETA,R,L,I,T,N1,                      GIS01190
C    1N3,BF,DLUDP                                                               GIS01200
C                                                                               GIS01210
C   START LENS CONSTRUCTION BY TRACING OUTERMOST RAY                            GIS01220
C                                                                               GIS01230
      FILTER=0                                                                  GIS01240
      J=1                                                                       GIS01250
C                                                                               GIS01260
CC    FIND OUTERMOST INSIDE SURFACE POINT                                       GIS01270
CC                                                                              GIS01280
      UDP(J)=BETA                                                               GIS01290
      X2(J)=(BF*DSIN(UDP(J)))/(DSIN(UDP(J))+(DTAN(ALPHA))*DCOS(UDP(J)))         GIS01300
      Y2(J)=(X2(J))*DTAN(ALPHA)                                                 GIS01310
C                                                                               GIS01320
CC    CALCULATE ALL LINE CONSTANTS                                              GIS01330
C                                                                               GIS01340
      I2P(J)=-ALPHA+DATAN((BF-X2(J))/Y2(J))                                     GIS01350
      RO=DSQRT((X2(J)+OSYMB)**2+Y2(J)**2)                                       GIS01360
      RZERO=RO                                                                  GIS01370
      IF(OSYMB.LT.-0.6) RZERO=DSQRT(R**2+OSYMB**2)                              GIS01380
      CALL INDEX(RO,N20,A,B,RZERO)                                              GIS01390
      I2(J)=DARSIN((N3/N20)*DSIN(I2P(J)))                                       GIS01400
      UP(J)=PI2-ALPHA-I2(J)                                                     GIS01410
      BASE=X2(J)+OSYMB                                                          GIS01420
      IF(BASE .EQ. 0.0) OSYMB=1.0000001*OSYMB                                   GIS01430
      BASE=X2(J)+OSYMB                                                          GIS01440
```

```
      THETAO=DATAN(Y2(J)/BASE)                                          GIS01450
      IF(THETAO .LT. 0.0)THETAO=PI+THETAO                               GIS01460
      ED=T/R                                                            GIS01470
      X1H=X2(J)-ED*DCOS(UP(J))                                          GIS01480
      Y1H=Y2(J)+ED*DSIN(UP(J))                                          GIS01490
      BAS1=X1H+OSYMB                                                    GIS01500
      IF(BAS1 .EQ. 0.0)OSYMB=1.0000001*OSYMB                            GIS01510
      BAS1=X1H+OSYMB                                                    GIS01520
      THETAE=DATAN(Y1H/BAS1)                                            GIS01530
      IF(THETAE .LT. 0.0)THETAE=PI+THETAE                               GIS01540
      PSIO=PI-UP(J)-THETAO                                              GIS01550
      EPSLON=+1.0                                                       GIS01560
      IF(PSIO .GT. PI2.OR.PSIO .LT. 0.0) EPSLON=-1.0                    GIS01570
      E=N2O*RO*DSIN(PSIO)                                               GIS01580
         ARG=DSQRT(A**2+4*B*(E**2)/(RZERO**2))                          GIS01590
         RSINO=DARSIN((2*(E**2)/(RO**2)-A)/ARG)                         GIS01600
         C=PI2-RSINO                                                    GIS01610
         IF(PSIO.LT.PI2) C=0.0                                          GIS01620
         TCRIT=THETAO-(EPSLON/2.)*(PI2-RSINO)                           GIS01630
         IF(THETAE.GE.TCRIT) THETE=THETAE-C                             GIS01640
         IF(THETAE.LT.TCRIT) THETE=THETAE                               GIS01650
         IF(THETAE.GE.TCRIT) EPSLON=+1.0                                GIS01660
         THETE=THETE-THETAO                                             GIS01670
C                                                                       GIS01680
CC    FIND RADIUS CORRESPONDING TO ANGLE SPECIFIED BY EDGE THICKNESS    GIS01690
CC    NOTE THAT NO ITERATION REQUIRED FOR FIRST RAY                     GIS01700
C                                                                       GIS01710
      CALL RADIUS(A,E,ARG,RSINO,THETE,RAD,RDOTG,FILTER,EPSLON)          GIS01720
      IF(FILTER .EQ. 1)WRITE(6,1800)                                    GIS01730
      IF(FILTER .EQ. 1)GO TO 82                                         GIS01740
C                                                                       GIS01750
CC    CALCULATE COORDINATES, ANGLE AND SLOPE FOR FRONT SURFACE SOLUTION  GIS01760
C                                                                       GIS01770
      X1(J)=RAD*COS(THETAE)-CSYMB                                       GIS01780
      Y1(J)=RAD*SIN(THETAE)                                             GIS01790
      CALL INDEX(RAD,N2,A,B,RZERO)                                      GIS01800
      PSI=DARSIN(E/(N2*RAD))                                            GIS01810
      IF(THETAE.LT.TCRIT) PSI=PI-PSI                                    GIS01820
      ZETA=PI-PSI-THETAE                                                GIS01830
      I1(J)=DARSIN(DSQRT(((DSIN(ZETA-U))**2)/((DCOS(ZETA-U)-(N1/N2))**2+GIS01840
     1(DSIN(ZETA-U))**2)))                                              GIS01850
      I1P(J)=DARSIN((N1/N2)*DSIN(I1(J)))                                GIS01860
      DYDXN(J)=-DTAN(I1(J)+U)                                           GIS01870
      DYDXT(J)=DCOTAN(I1(J)+U)                                          GIS01880
      IF(SHAPE .EQ. 0) GO TO 5                                          GIS01890
      WRITE(6,200) J,X1(J),Y1(J),X2(J),Y2(J),UOP(J),I2P(J),I2(J),UP(J), GIS01900
     1I1P(J),I1(J),DYDXT(J)                                             GIS01910
    5 CONTINUE                                                          GIS01920
```

```
C                                                                       GIS01930
CC    END OF TRACE FOR 1ST RAY                                          GIS01940
CC                                                                      GIS01950
CC    START ITERATIVE SOLUTION SCHEME                                   GIS01960
C                                                                       GIS01970
      K=I+1                                                             GIS01980
      SURFL=0.0                                                         GIS01990
      DO  10  J=2,K                                                     GIS02000
          TRIP=0                                                        GIS02010
          L=J-1                                                         GIS02020
C                                                                       GIS02030
CC     CALCULATE INNER SURFACE COORDINATES                              GIS02040
C                                                                       GIS02050
          UDP(J)=BETA-(L)*DLUDP                                         GIS02060
          X2(J)=(BF*DSIN(UDP(J)))/(DSIN(UDP(J))+(DTAN(ALPHA))*         GIS02070
     1    DCOS(UDP(J)))                                                 GIS02080
          Y2(J)=(X2(J))*DTAN(ALPHA)                                     GIS02090
C                                                                       GIS02100
CC      CALCULATE ALL LINE CONSTANTS                                    GIS02110
C                                                                       GIS02120
          RO=DSQRT((X2(J)+OSYMB)**2+Y2(J)**2)                           GIS02130
          IF(J .EQ. K) ROOO=RO                                          GIS02140
          I2P(J)=-ALPHA+DATAN((BF-X2(J))/Y2(J))                         GIS02150
          CALL INDEX(RO,N20,A,B,RZERO)                                  GIS02160
          I2(J)=DARSIN((N3/N20)*DSIN(I2P(J)))                           GIS02170
          LP(J)=PI2-ALPHA-I2(J)                                         GIS02180
          THETAO=DATAN(Y2(J)/(X2(J)+OSYMB))                             GIS02190
          IF(THETAO.LT.0.0) THETAO=PI+THETAO                            GIS02200
          PSIO=PI-LP(J)-THETAO                                          GIS02210
          EPSLON=1.0                                                    GIS02220
          IF(PSIO .GT. PI2.OR.PSIO.LT.0.0) EPSLON=-1.0                  GIS02230
          E=N2O*RO*DSIN(PSIO)                                           GIS02240
          Y1H=((DCOTAN(I1(L)+U))*(DCOTAN(LP(J)))*(Y2(J)+               GIS02250
     1    X2(J)*DTAN(UP(J)))+Y1(L)-X1(L)*DCOTAN(I1(L)+U))/(1.0+         GIS02260
     1    (DCOTAN(I1(L)+U))*DCOTAN(UP(J)))                              GIS02270
          X1H=(DCOTAN(UP(J)))*(-Y1H+Y2(J)+X2(J)*DTAN(UP(J)))            GIS02280
          IF(B .EQ. 0.0) GO TO 8                                        GIS02290
          ARG=DSQRT(A**2+4*B*(E**2)/(RZERO**2))                         GIS02300
          RSINO=DARSIN((2*(E**2)/(RO**2)-A)/ARG)                        GIS02310
          C=PI2-RSINO                                                   GIS02320
          IF(PSIO.LT.PI2) C=0.0                                         GIS02330
          TCRIT=THETAO-(EPSLON/2.)*(PI2-RSINO)                          GIS02340
          THETAP=DATAN(Y1H/(X1H+OSYMB))                                 GIS02350
          IF(THETAP.LT.0.0) THETAP=PI+THETAP                            GIS02360
          RP=DSQRT((X1H+OSYMB)**2+Y1H**2)                               GIS02370
          SL=DCOTAN(I1(L)+L)                                            GIS02380
          BB=Y1H-SL*(X1H+OSYMB)                                         GIS02390
C                                                                       GIS02400
```

```
C      IF ANGLE LARGE ENOUGH GO TO ANGLE IN TERMS OF RADIUS ITERATION      GIS02410
C                                                                          GIS02420
          IF(THETA0.GT.PI2) GO TO 106                                      GIS02430
C                                                                          GIS02440
C         NEWTON-RAPHSON ITERATION                                         GIS02450
C                                                                          GIS02460
    6     CONTINUE                                                         GIS02470
          TRIP=TRIP+1                                                      GIS02480
          IF(THETAP.GE.TCRIT.AND.OSYMB.LT.0.0) THETP=THETAP-C              GIS02490
          IF(THETAP.LT.TCRIT.AND.OSYMB.LT.0.0) THETP=THETAP                GIS02500
          IF(THETAP.GE.TCRIT.AND.OSYMB.LT.0.0) EPSLON=+1.0                 GIS02510
          IF(OSYMB.GE.0.0) THETP=THETAP                                    GIS02520
          THETAH=THETP-THETA0                                              GIS02530
          CALL RADYUS(A,E,ARG,RSINO,THETAH,RAD,RDG,FILTER,EPSLON)          GIS02540
          IF(FILTER .EQ. 1) WRITE(6,1800)                                  GIS02550
          IF(FILTER .EQ. 1) GO TO 82                                       GIS02560
          CALL RADUS(PI,X1(L),Y1(L),I1(L),L,OSYMB,THETAP,RADD,RDGG)        GIS02570
          RDF=RAD-RADD                                                     GIS02580
          RDT=RDG-RDGG                                                     GIS02590
          FACTOR=1.7/(DABS(PSIO))**.5                                      GIS02600
          IF(OSYMB.GT.0.0) FACTOR=1.3                                      GIS02610
          THETAN=THETAP-RDF/(RDT*FACTOR)                                   GIS02620
          DIFF=DABS(THETAN-THETAP)                                         GIS02630
          EXIT=0.000001*DABS(PSIO)                                         GIS02640
          IF(OSYMB.GT.0.0) EXIT=0.0000001                                  GIS02650
          EXIT2=300                                                        GIS02660
          IF(DIFF.LT.EXIT) GO TO 7                                         GIS02670
          IF(TRIP.GT.EXIT2) GO TO 7                                        GIS02680
          THETAP=THETAN                                                    GIS02690
          GO TO 6                                                          GIS02700
C                                                                          GIS02710
C     ANGLE IN TERMS OF RADIUS ITERATION                                   GIS02720
C                                                                          GIS02730
  106     TRIP=TRIP+1                                                      GIS02740
          RSIN=DARSIN((2*E**2/RP**2-A)/ARG)                                GIS02750
          THETA=THETA0-EPSLON/2*(RSIN-RSINO)                               GIS02760
          CHECK1=THETA-THETAP                                              GIS02770
          RPP=EE/(DSIN(THETA)-SL*DCOS(THETA))                              GIS02780
          CHECK2=DABS(RPP-RP)                                              GIS02790
          THETAP=THETA                                                     GIS02800
          RP=RPP                                                           GIS02810
          IF(TRIP.GT.100) GO TO 107                                        GIS02820
          IF(CHECK2.LT.0.0000001) GO TO 107                                GIS02830
          GO TO 106                                                        GIS02840
  107     RAD=RPP                                                          GIS02850
C                                                                          GIS02860
C    FIND COORDINATES, ANGLE AND SLOPE OF FRONT SURFACE FOR SOLUTION       GIS02870
C                                                                          GIS02880
```

```
    7       X1(J)=RAD*DCOS(THETAP)-OSYMB                                    GIS02890
            Y1(J)=RAD*DSIN(THETAP)                                          GIS02900
            CALL INDEX(RAD,N2,A,B,RZERO)                                    GIS02910
            EONR=E/(N2*RAD)                                                 GIS02920
            IF(ECNR .GT. 1.0) WRITE(6,9978)ECNR                             GIS02930
 9978       FORMAT(10X,'EONR =',F19.16)                                     GIS02940
            IF(ECNR .GT. 1.0)ECNR=1.0                                       GIS02950
            PSIP=DARSIN(EONR)                                               GIS02960
            IF(THETAP.LT.TCRIT.AND.PSIO.GT.0.0) PSIP=PI-PSIP                GIS02970
            ZETA=PI-PSIP-THETAP                                             GIS02980
            GO TO 9                                                         GIS02990
C                                                                           GIS03000
    8   CONTINUE                                                            GIS03010
            X1(J)=X1H                                                       GIS03020
            Y1(J)=Y1H                                                       GIS03030
            ZETA=UP(J)                                                      GIS03040
C                                                                           GIS03050
    9   CONTINUE                                                            GIS03060
            IF(Y1(J) .LE. 0.0) GO TO 11                                     GIS03070
            I1(J)=DARSIN(DSQRT((DSIN(ZETA-U))**2/((DCOS(ZETA-U)-(N1/N2))**2GIS03080
     1      +(DSIN(ZETA-U))**2)))                                           GIS03090
            I1P(J)=DARSIN((N1/N2)*DSIN(I1(J)))                              GIS03100
            DYDXN(J)=(-DTAN(I1(J)+U))                                       GIS03110
            DYDXT(J)=DCOTAN(I1(J)+U)                                        GIS03120
          SLINCR=DSQRT((X1(J)-X1(L))**2+(Y1(L)-Y1(J))**2)                   GIS03130
          SURFL=SURFL+SLINCR                                                GIS03140
            IF(SHAPE .EQ. 0) GO TO 10                                       GIS03150
            WRITE(6,200)J,X1(J),Y1(J),X2(J),Y2(J),UDF(J),I2P(J),I2(J),      GIS03160
     1      UP(J),I1P(J),I1(J),DYDXT(J)                                     GIS03170
   10   CONTINUE                                                            GIS03180
        GO TO 12                                                            GIS03190
C                                                                           GIS03200
C     END OF TRACE FOR FRONT SURFACE CONSTRUCTION                           GIS03210
C                                                                           GIS03220
C     OPAQUE NOSE SECTION CALCULATIONS                                      GIS03230
C                                                                           GIS03240
   11   CONTINUE                                                            GIS03250
        K=J-1                                                               GIS03260
   12   CONTINUE                                                            GIS03270
        AB=(-X1(K))+(Y1(K))/DYDXT(K)                                        GIS03280
        STATNA=-AB                                                          GIS03290
        GAMMA=ATAN(DYDXT(K))                                                GIS03300
        ONP=(Y1(K))/SIN(GAMMA)                                              GIS03310
        TSURFL=ONP+SURFL                                                    GIS03320
        WRITE(6,300) GAMMA,STATNA,ONP,SURFL,TSURFL                          GIS03330
C                                                                           GIS03340
C       WRITE OUTPUT FILE FOR PROGRAM CHECK                                 GIS03350
C                                                                           GIS03360
```

```
      WRITE(2,110)A,B,CSYMB,I,ALPHA,N3,F,R                                  GISO3370
      DO 120 J=1,I                                                          GISO3380
  120 WRITE(2,111)X1(J),Y1(J),X2(J),Y2(J)                                   GISO3390
      WRITE(2,112)                                                          GISO3400
  110 FORMAT(' ',3F11.7,I5,F10.7,3F5.3)                                     GISO3410
  111 FORMAT(' ',4F10.7)                                                    GISO3420
  112 FORMAT(' ','/*')                                                      GISO3430
C                                                                           GISO3440
C    LENS INDEX OF REFRACTION DATA -- N2(R) --                              GISO3450
C                                                                           GISO3460
      BEFF=B/(RZERO**2)                                                     GISO3470
      SQRE=A+BEFF*RZERO**2                                                  GISO3480
      IF(CSYMB.LT.0.0) SQRE=A+BEFF*(RZERO/1.5)**2                           GISO3490
      N2X21=DSQRT(SQRE)                                                     GISO3500
      CALL INDEX(RAD,N2X1K,A,B,RZERO)                                       GISO3510
      CALL INDEX(ROOO,N2X2K,A,B,RZERO)                                      GISO3520
      PRCNTE=-((N2X2K-N2X21)/N2X2K)*100.0                                   GISO3530
      PRCNTT=-((N2X2K-N2X1K)/N2X2K)*100.0                                   GISO3540
      PRCNTO=-((N2X1K-N2X21)/N2X1K)*100.0                                   GISO3550
      WRITE(6,1700)A,BEFF,CSYMB,PRCNTE,PRCNTT,PRCNTO                        GISO3560
C                                                                           GISO3570
C ***********************************************************************  GISO3580
C                                                                           GISO3590
C    SECTION 2. SKEW RAY TRACE                                              GISO3600
C                                                                           GISO3610
C ***********************************************************************  GISO3620
C                                                                           GISO3630
C                                                                           GISO3640
C  INITIALIZE                                                               GISO3650
C                                                                           GISO3660
      G=1                                                                   GISO3670
      COUNT=0                                                               GISO3680
      COUNT2=0                                                              GISO3690
      RAYY(G)=0.0                                                           GISO3700
      RAYZ(G)=0.0                                                           GISO3710
      NTANCTY(G)=1.0                                                        GISO3720
      XMAVET=0.0                                                            GISO3730
      XMAVE=0.0                                                             GISO3740
      IF(SQURAY .EQ. 1) WRITE(6,400) ALFAP,GRID                             GISO3750
C                                                                           GISO3760
C   CONSTRUCT GRID                                                          GISO3770
C                                                                           GISO3780
   20 CONTINUE                                                              GISO3790
      IF(RAYZ(G) .GE. Y1(K)) GO TO 30                                       GISO3800
        CENI=(AB+X1(K))*SIN(ALFAP)                                          GISO3810
        TORBI=DSQRT(Y1(K)**2-RAYZ(G)**2)                                    GISO3820
        BOTI=CENI-TORBI                                                     GISO3830
        IF(RAYY(G) .LE. BOTI) GO TO 30                                      GISO3840
```

```
            TOPI=CENI+TORBI                                                    GIS03850
            IF(RAYY(G) .GE. TOPI) GO TO 30                                     GIS03860
              RAYZ(G)=RAYZ(G)+GRID                                             GIS03870
              GO TO 20                                                         GIS03880
     30 CONTINUE                                                               GIS03890
C    SEARCH FOR Y1(J) JUST GT RAYZ(G); RETRIEVE X1(J)                          GIS03900
        M=K/2                                                                  GIS03910
        IF(Y1(M) .LT. RAYZ(G)) GO TO 32                                        GIS03920
          IF(RAYY(G) .GT. 1.15) GO TO 32                                       GIS03930
            N=K-1                                                              GIS03940
     31 CONTINUE                                                               GIS03950
        IF(Y1(N) .GT. RAYZ(G)) GO TO 36                                        GIS03960
          N=N-1                                                                GIS03970
          IF(N .EQ. 0) GO TO 34                                                GIS03980
          GO TO 31                                                             GIS03990
     32 CONTINUE                                                               GIS04000
        N=M                                                                    GIS04010
     33 CONTINUE                                                               GIS04020
        IF(Y1(N) .GT. RAYZ(G)) GO TO 36                                        GIS04030
          N=N-1                                                                GIS04040
          IF(N .EQ. 0) GO TO 34                                                GIS04050
            GO TO 33                                                           GIS04060
     34 CONTINUE                                                               GIS04070
        RAYZ(G)=0.0                                                            GIS04080
        IF(RAYY(G) .LT. 0.0) GO TO 35                                          GIS04090
          RAYY(G)=RAYY(G)+GRID                                                 GIS04100
          GO TO 20                                                             GIS04110
     35 CONTINUE                                                               GIS04120
        RAYY(G)=RAYY(G)-GRID                                                   GIS04130
        GO TO 20                                                               GIS04140
     36 CONTINUE                                                               GIS04150
        RADIUS=DSQRT(RAYZ(G)**2+((RAYY(G)/COS(ALFAP))-(X1(N)+AB)*              GIS04160
       1TAN(ALFAP))**2)                                                        GIS04170
        IF(Y1(N) .GT. RADIUS) GO TO 40                                         GIS04180
          IF(RADIUS .GT.Y1(1)) GO TO 37                                        GIS04190
          N=N-1                                                                GIS04200
          GO TO 36                                                             GIS04210
     37 CONTINUE                                                               GIS04220
        N=N-1                                                                  GIS04230
        IF(N .GT. 0) GO TO 36                                                  GIS04240
        CEN=(X1(1)+AB)*SIN(ALFAP)                                              GIS04250
        TORB=Y1(1)*COS(ALFAP)                                                  GIS04260
        RAYZ(G)=0.0                                                            GIS04270
        IF(RAYY(G) .GE. 0.0) GO TO 38                                          GIS04280
          RAYY(G)=RAYY(G)-GRID                                                 GIS04290
          BOT=CEN-TORB                                                         GIS04300
C         CHECK FOR BOTTOM EDGE OF LENS, GO TO MIRROR IMAGE RAYS               GIS04310
          IF(RAYY(G) .LT. BOT) GO TO 60                                        GIS04320
```

```
          GO TO 20                                                                    GIS04330
   38 CONTINUE                                                                        GIS04340
      RAYY(G)=RAYY(G)+GRID                                                            GIS04350
      TOP=CEN+TORE                                                                    GIS04360
      IF(RAYY(G) .GT. TOP) GO TO 39                                                   GIS04370
        GO TO 20                                                                      GIS04380
   39 CONTINUE                                                                        GIS04390
      RAYY(G)=-GRID                                                                   GIS04400
      GO TO 20                                                                        GIS04410
   40 CONTINUE                                                                        GIS04420
      P=N+1                                                                           GIS04430
C                                                                                     GIS04440
C     CALCULATE INTERCEPT POINT ON OUTER SURFACE                                      GIS04450
C                                                                                     GIS04460
      AO=RAYY(G)/COS(ALFAP)                                                           GIS04470
      BO=TAN(ALFAP)                                                                   GIS04480
      CO=(Y1(N)-Y1(P))/(X1(N)-X1(P))                                                  GIS04490
      DO=X1(P)                                                                        GIS04500
      EO=Y1(P)                                                                        GIS04510
      PAR1=(AC*BO+CO*EO-(CO**2)*DO-AB*(BC**2))                                        GIS04520
      PAR2=(EC**2-CO**2)                                                              GIS04530
      PAR3=(AC**2+AB*(BO**2)*AB-2.0*AO*BC*AB+                                         GIS04540
     1RAYZ(G)**2-(EO-CO*DO)**2)                                                       GIS04550
      XO=(PAR1/PAR2)+SQRT((PAR1/PAR2)**2-PAR3/PAR2)                                   GIS04560
      IF((PAR1/PAR2).GT.X1(1))XO=XO-2.0*SQRT((PAR1/PAR2)**2-PAR3/PAR2)                GIS04570
      YO=(RAYY(G)/COS(ALFAP))-(XO+AB)*TAN(ALFAP)                                      GIS04580
      ZO=RAYZ(G)                                                                      GIS04590
C                                                                                     GIS04600
C     CALCULATE THE DIRECTION COSINES OF OUTSIDE SURFACE NORMAL                       GIS04610
C                                                                                     GIS04620
      DELTAX=X1(N)-X1(P)                                                              GIS04630
      DELTAY=Y1(N)-Y1(P)                                                              GIS04640
      SP=DSQRT((RADIUS-Y1(P))**2+(XO-X1(P))**2)                                       GIS04650
      ST=SQRT(DELTAX**2+DELTAY**2)                                                    GIS04660
      RATIO=SP/ST                                                                     GIS04670
      DYDXNP=RATIO*(DYDXN(N)-DYDXN(P))+DYDXN(P)                                       GIS04680
      NPI=1.0/DYDXNP                                                                  GIS04690
      IF(YO .EQ. 0.0) GO TO 41                                                        GIS04700
        ZOYO=(ZO/YO)                                                                  GIS04710
        IF(ZOYO .GE. 8235000) GO TO 41                                                GIS04720
        IF(ZOYO .LE. -8235000) GO TO 42                                               GIS04730
          NPJ=DCOS(DATAN(ZO/YO))                                                      GIS04740
          NPK=DSIN(DATAN(ZO/YO))                                                      GIS04750
          GO TO 43                                                                    GIS04760
   41 CONTINUE                                                                        GIS04770
      NPJ=0.0                                                                         GIS04780
      NPK=1.0                                                                         GIS04790
      GO TO 43                                                                        GIS04800
```

```
   42 CONTINUE                                                          GIS04810
      NPJ=0.0                                                           GIS04820
      NPK=-1.0                                                          GIS04830
   43 CONTINUE                                                          GIS04840
C                                                                       GIS04850
CC    DIRECTION COSINES OF EXTERNAL RAY                                 GIS04860
C                                                                       GIS04870
      CK=COS(ALFAP)                                                     GIS04880
      CL=-SIN(ALFAP)                                                    GIS04890
C                                                                       GIS04900
CC    DIRECTION COSINES OF OUTSIDE SURFACE NORMAL                       GIS04910
C                                                                       GIS04920
      LK=NPI/(SQRT(NPI**2+NPJ**2+NPK**2))                               GIS04930
      LL=(NPJ*LK)/NPI                                                   GIS04940
      LM=(NPK*LK)/NPI                                                   GIS04950
      IF(YO .LT. 0.0) GO TO 44                                          GIS04960
        GO TO 45                                                        GIS04970
   44 CONTINUE                                                          GIS04980
      LL=-LL                                                            GIS04990
      LM=-LM                                                            GIS05000
   45 CONTINUE                                                          GIS05010
      THETA=DARCOS(CK*LK+CL*LL)                                         GIS05020
C                                                                       GIS05030
CC    ANGLES OF INCIDENCE AND REFRACTION OUTSIDE SURFACE                GIS05040
C                                                                       GIS05050
      PHI=PI-THETA                                                      GIS05060
      RO=DSQRT((XO+OSYMB)**2+YO**2+ZO**2)                               GIS05070
      CALL INDEX(RO,N20,A,B,RZERO)                                      GIS05080
      ART=(N1/N20)*DSIN(PHI)                                            GIS05090
      IF(ART.GT.1.0) ART=1.0                                            GIS05100
      IF(ART.LT.-1.0) ART=-1.0                                          GIS05110
      PHIP=DARSIN(ART)                                                  GIS05120
C                                                                       GIS05130
CC    CALCULATION OF TRANSMITED INTENSITY AT OUTSIDE SURFACE            GIS05140
C                                                                       GIS05150
      FACE=1                                                            GIS05160
      CALL XMIT(PHI,N20,XMTNO,FACE,XM,N1,N3)                            GIS05170
      IF(FACE .EQ. 3) GO TO 53                                          GIS05180
      NUM=DCOS(PHIP)-(N1/N20)*DCOS(PHI)                                 GIS05190
C                                                                       GIS05200
CC    INITIAL DIRECTION COSINES OF INTERNAL REFRACTED RAY               GIS05210
C                                                                       GIS05220
      CKP=(N1/N20)*CK-NUM*LK                                            GIS05230
      CLP=(N1/N20)*CL-NUM*LL                                            GIS05240
      CMP=-NUM*LM                                                       GIS05250
      CKKP=CKP                                                          GIS05260
      CLLP=CLP                                                          GIS05270
      CMMP=CMP                                                          GIS05280
```

```
C                                                                       GISO5290
C       INTERNAL RAY LENGTH AND INSIDE SURFACE INTERCEPT                GISO5300
C                                                                       GISO5310
        ROX=(XC+OSYMB)/RO                                               GISO5320
        ROY=YC/RC                                                       GISO5330
        ROZ=ZO/RC                                                       GISO5340
        NPOX=RCY*CMP-ROZ*CLP                                            GISO5350
        NPOY=RCZ*CKP-ROX*CMP                                            GISO5360
        NPOZ=RCX*CLP-ROY*CKP                                            GISO5370
        NPO=DSQRT(NPOX**2+NPOY**2+NPOZ**2)                              GISO5380
        NPOX=NPCX/NPO                                                   GISO5390
        NPOY=NPCY/NPO                                                   GISO5400
        NPOZ=NPCZ/NPO                                                   GISO5410
C                                                                       GISO5420
C   CALCULATE PSIO AND THEN DETERMINE ITS SIGN                          GISO5430
C                                                                       GISO5440
        PSIO=DARCOS(ROX*CKP+ROY*CLP+ROZ*CMP)                            GISO5450
        IF(NPOZ.EQ.0.0) XX=OSYMB*ROZ/(ROX*NPOY)*(NPCY*ROY/ROZ+NPOZ)     GISO5460
        IF(NPOZ.EQ.0.0) GO TO 121                                       GISO5470
        XX=OSYMB*ROY/(ROX*NPOZ)*(NPOZ*ROZ/RCY+NPOY)                     GISO5480
  121 CONTINUE                                                          GISO5490
        CCK=OSYMB/DSQRT(OSYMB**2+XX**2)                                 GISO5500
        CCL=0.0                                                         GISO5510
        CCM=XX/DSQRT(OSYMB**2+XX**2)                                    GISO5520
        IF(NPOZ.EQ.0.0) CCL=XX/DSQRT(OSYMB**2+XX**2)                    GISO5530
        IF(NPOZ.EQ.0.0) CCM=0.0                                         GISO5540
        THETA1=PI-DARCOS(CCK*CKP+CCL*CLP+CCM*CMP)                       GISO5550
        THETA2=DARCOS(CCK*ROX+CCL*ROY+CCM*ROZ)                          GISO5560
        FLUG=0                                                          GISO5570
        IF(THETA1.GT.THETA2.AND.OSYMB.LT.0.0) FLUG=1                    GISO5580
        IF(FLUG.EQ.1) PSIO=-PSIO                                        GISO5590
C                                                                       GISO5600
C   CALCULATE LINE CONSTANTS                                            GISO5610
C                                                                       GISO5620
        EPSLON=+1.0                                                     GISO5630
        IF(PSIO.LT.PI.AND.PSIO.GT.PI2) EPSLON=-1.0                      GISO5640
        E=N20*RO*DSIN(PSIO)                                             GISO5650
        ARG=DSQRT(A**2+4*B*E**2/RZERO**2)                               GISO5660
        RSINO=DARSIN((2*E**2/RO**2-A)/ARG)                              GISO5670
        C=PI2-RSINO                                                     GISO5680
        IF(PSIO.LT.PI2) C=0.0                                           GISO5690
        TCRIT=-EPSLON/2*(PI2-RSINO)                                     GISO5700
        NUM1=(CLP*YC+CMP*ZC-CKP*XO*(DTAN(ALPHA)**2))                    GISO5710
        NUM2=(CLP**2+CMP**2-(CKP**2)*(DTAN(ALPHA)**2))                  GISO5720
        NUM3=(YC**2+ZO**2-(XO**2)*(DTAN(ALPHA)**2))                     GISO5730
        NUM4=NUM1**2-NUM2*NUM3                                          GISO5740
C                                                                       GISO5750
C       CHECK FOR INTERCEPT WITH INSIDE SURFACE                         GISO5760
```

```
C                                                                       GIS05770
      IF(NUM4 .LT. 0.0) GO TO 50                                        GIS05780
C                                                                       GIS05790
C     CALCULATE FIRST GUESS POINT                                       GIS05800
C                                                                       GIS05810
        D2=-NUM1/NUM2-DSQRT(NUM4)/NUM2                                  GIS05820
          XIH=D2*CKP+XO                                                 GIS05830
          YIH=D2*CLP+YO                                                 GIS05840
          ZIH=D2*CMP+ZO                                                 GIS05850
          IF(ZIH .LT. 0.0) GO TO 50                                     GIS05860
          OSYMPH=DSQRT((CSYMB+XIH)**2+YIH**2+ZIH**2)                    GIS05870
          OPHX=(CSYMB+XIH)/CSYMPH                                       GIS05880
          OPHY=YIH/OSYMPH                                               GIS05890
          OPHZ=ZIH/OSYMPH                                               GIS05900
          THETAP=ARCOS(ROX*OPHX+ROY*OPHY+ROZ*OPHZ)                      GIS05910
          RP=CSYMPH                                                     GIS05920
          IF(E.EQ.0.0) GO TO 124                                        GIS05930
          SIGN=-1.0                                                     GIS05940
          IF(CSYMB .GT. 0.0) SIGN=+1.0                                  GIS05950
          IF(PSIO.LT.0.0) THETAP=-THETAP                                GIS05960
C                                                                       GIS05970
C  INTIALIZE SURF ROUTINE TO FIND CORRECT VALUE OF SIGN                 GIS05980
C                                                                       GIS05990
       CALL SURF(THETAP,RADD,RDOTS,SIGN,FILTER,NPOX,NPOY,NPOZ,PSIO,     GIS06000
      1XO,YO,ZO,OSYMB,RO,CKKP,CLLP,CMMP,ALPHA)                          GIS06010
          IF(RADD .LT. 0.0) SIGN=-SIGN                                  GIS06020
          THETP=PI-THETA+THETAP                                         GIS06030
C                                                                       GIS06040
C    CALL NEWTON-RAPHSON ITERATION IF ANGLE LARGE ENOUGH                GIS06050
C                                                                       GIS06060
          IF(CABS(PSIO).GT.3.0.AND.OSYMB.LT.0.0)                        GIS06070
     1    CALL INTER2(RP,THETAP,RAD,THTAP,A,E,ARG,RSINO,EPSLON,SIGN,    GIS06080
     1    NPOX,NPOY,NPOZ,PSIO,XO,YO,ZO,OSYMB,RO,CKKP,CLLP,CMMP,         GIS06090
     1    ALPHA,THETA,FILTER,FLUG)                                      GIS06100
          IF(CABS(PSIO).GT.3.0.AND.OSYMB.LT.0.0) GO TO 118              GIS06110
C                                                                       GIS06120
C    CALL ANGLE IN TERMS OF RADIUS ROUTINE IN ANGLE SMALL ENOUGH        GIS06130
C                                                                       GIS06140
          CALL INTER1(SIGN,THETAP,RAD,THTAP,FILTER,A,E,ARG,RSINO,       GIS06150
     1EPSLON,C,TCRIT,NPOX,NPOY,NPOZ,PSIO,XO,YO,ZO,RO,CKKP,CLLP,CMMP,    GIS06160
     1ALPHA,OSYMB,FLUG)                                                 GIS06170
  118     CONTINUE                                                      GIS06180
C                                                                       GIS06190
C    CHECK FOR FAILED RAYS                                              GIS06200
C                                                                       GIS06210
          IF(FILTER .EQ. 2 .OR. FILTER .EQ. 1) GO TO 51                 GIS06220
          IF(FILTER .EQ. 3) GO TO 50                                    GIS06230
          IF(FILTER .EQ. 4) GO TO 56                                    GIS06240
```

```
C                                                                       GISO6250
      GO TO 123                                                         GISO6260
 124  THTAP=THETAP                                                      GISO6270
      RAD=RP                                                            GISO6280
C                                                                       GISO6290
C   CALCULATE ALL REQUIRED PARAMETERS AT SOLUTION POINT                 GISO6300
C                                                                       GISO6310
 123  BETAO=DSIN(THTAP)/DSIN(PSIO)                                      GISO6320
      ALFAO=DCOS(THTAP)-BETAO*DCOS(PSIO)                                GISO6330
      XI=RAD*(ALFAO*((XO+OSYMB)/RO)+BETAO*CKKP)-OSYMB                   GISO6340
      YI=RAD*(ALFAO*(YO/RO)+BETAO*CLLP)                                 GISO6350
      ZI=RAD*(ALFAO*(ZO/RO)+BETAO*CMMP)                                 GISO6360
      CALL INDEX(RAD,N2,A,B,RZERO)                                      GISO6370
      EONR=DABS(E/(N2*RAD))                                             GISO6380
      IF(EONR .GT. 1.0) WRITE(6,9978)EONR                               GISO6390
      IF(EONR .GT. 1.0) EONR=1.0                                        GISO6400
      PSIP=DARSIN(EONR)                                                 GISO6410
      IF(CSYMB.LT.0.0.AND.TCRIT.GT.THTAP) PSIP=PI-PSIP                  GISO6420
C                                                                       GISO6430
C   CALCULATE REFRACTION OF RAY AT INNER SURFACE                        GISO6440
C                                                                       GISO6450
      A1=(XI+CSYMB)/RAD                                                 GISO6460
      B1=YI/RAD                                                         GISO6470
      C1=ZI/RAD                                                         GISO6480
      CALL DIRECT(CKP,CLP,CMP,DIR,A1,B1,C1,NPOX,NPOY,NPOZ,PSIP)         GISO6490
    IF(XI .LT. 0.0 .OR. XI .GT. X2(1)) GO TO 51                         GISO6500
      NUM5=DSQRT((XI**2)*(DTAN(ALPHA)**4)+YI**2+ZI**2)                  GISO6510
C                                                                       GISO6520
C     DIRECTION COSINES OF INSIDE NORMAL                                GISO6530
C                                                                       GISO6540
      LKP=-XI*(DTAN(ALPHA)**2)/NUM5                                     GISO6550
      LLP=YI/NUM5                                                       GISO6560
      LMP=ZI/NUM5                                                       GISO6570
      KKLLMM=CKP*LKP+CLP*LLP+CMP*LMP                                    GISO6580
      IF(KKLLMM .GE. 1.0)THETAI=0.0                                     GISO6590
      IF(KKLLMM .LE. -1.0)THETAI=PI                                     GISO6600
      IF((DABS(KKLLMM)) .GE. 1.0) GO TO 457                             GISO6610
      THETAI=DARCOS((CKP*LKP)+(CLP*LLP)+(CMP*LMP))                      GISO6620
 457  CONTINUE                                                          GISO6630
C                                                                       GISO6640
C     ANGLES OF INCIDENCE AND TOTAL INTERNAL REFLECTION                 GISO6650
C     AT THE INSIDE SURFACE:                                            GISO6660
C                                                                       GISO6670
      PHII=PI-THETAI                                                    GISO6680
      IF(FLLG.EQ.1) PHII=THETAI                                         GISO6690
      TIRA=DARSIN(N3/N2)                                                GISO6700
C                                                                       GISO6710
C     CHECK FOR TOTAL INTERNAL REFLECTION:                              GISO6720
```

```
C                                                                       GISO6730
            IF(PHII .GE. TIRA) GO TO 52                                 GISO6740
              PHIIP=DARSIN((N2/N3)*DSIN(PHII))                          GISO6750
C                                                                       GISO6760
CC            CALCULATION OF TRANSMITTED INTENSITY AT INSIDE SURFACE    GISO6770
C                                                                       GISO6780
              FACE=2                                                    GISO6790
              CALL XMIT(PHII,N2,XMTNCP,FACE,XM,N1,N3)                   GISO6800
              NTNCTY(G)=XMTNC*XMTNCP                                    GISO6810
              XMAVET=XMAVET+NTNCTY(G)                                   GISO6820
              XMAVE=XMAVET/(G*1.0)                                      GISO6830
              NUM8=DCOS(PHIIP)-(N2/N3)*DCOS(PHII)                       GISO6840
C                                                                       GISO6850
CC    CALCULATE DIRECTION COSINES OF INSIDE EXTERNAL REFRACTED RAY      GISO6860
CC    CORRECTING FOR RAYS PAST REGION OF NEAR-RADIAL LINES              GISO6870
C                                                                       GISO6880
            IF(FLUG.EQ.1) CKP=-CKP                                      GISO6890
            IF(FLUG.EQ.1) CLP=-CLP                                      GISO6900
            IF(FLUG.EQ.1) CMP=-CMP                                      GISO6910
              CKPP=(N2/N3)*CKP-NUM8*LKP                                 GISO6920
              CLPP=(N2/N3)*CLP-NUM8*LLP                                 GISO6930
              CMPP=(N2/N3)*CMP-NUM8*LMP                                 GISO6940
C                                                                       GISO6950
CC    WRITE OUTPUT FOR PROGRAM DETPLOT                                  GISO6960
C                                                                       GISO6970
              WRITE(4,126) XI,YI,ZI,CKPP,CLPP,CMPP                      GISO6980
  126         FORMAT(' ',6F10.7)                                        GISO6990
C                                                                       GISO7000
CC            LENGTH OF INSIDE EXTERNAL REFRACTED RAY                   GISO7010
C                                                                       GISO7020
              D3=(BF-XI)/CKPP                                           GISO7030
              NUM6=((XO+AB)*CK+YO*CL)                                   GISO7040
              NUM7=((XO+AB)*(-CL)+YO*CK)                                GISO7050
C                                                                       GISO7060
CC            LENGTH OF OUTSIDE INCIDENT RAY                            GISO7070
C                                                                       GISO7080
              D1=DSQRT((XO-NUM6)**2+(YC-NUM7)**2)                       GISO7090
C                                                                       GISO7100
CC            TOTAL OPTICAL PATHLENGTH:                                 GISO7110
C                                                                       GISO7120
              OPL(G)=N1*D1+N2*D2+N3*D3                                  GISO7130
C                                                                       GISO7140
CC            INTERSECTION WITH THE IMAGE PLANE:                        GISO7150
C                                                                       GISO7160
              YIM(G)=((BF-XI)/CKPP)*CLPP+YI                             GISO7170
              ZIM(G)=((BF-XI)/CKPP)*CMPP+ZI                             GISO7180
              DIM(G)=DSQRT(YIM(G)**2+ZIM(G)**2)                         GISO7190
              IF(DABS(YIM(G)).GT.1.0.OR.DABS(ZIM(G)).GT.1.0.OR.DIM(G).GT. GISO7200
```

```
     1 1.0)    GO TO 55                                                          GIS07210
               IF(CMPP .LT. 0.000000000001 .AND. CMPP .GT. -0.000000000001)GIS07220
     1           GO TO 46                                                        GIS07230
                 XDIAPT(G)=(-ZIM(G)/CMPP)*CKPP+AE                                GIS07240
                 YDIAPT(G)=(-ZIM(G)/CMPP)*CLPP+YIM(G)                            GIS07250
                 GO TO 47                                                        GIS07260
    46 CONTINUE                                                                  GIS07270
               XDIAPT(G)=99999.                                                  GIS07280
               YDIAPT(G)=99999.                                                  GIS07290
    47 CONTINUE                                                                  GIS07300
               IF(SQURAY .EQ. 1) WRITE(6,500)G,RAYY(G),RAYZ(G),XO,YO,XI,         GIS07310
     1         YI,ZI,OPL(G),YIM(G),ZIM(G),NTACTY(G),XDIAPT(G),YDIAPT(G)         GIS07320
               FLAG(G)=0                                                         GIS07330
    49 CONTINUE                                                                  GIS07340
       H=G                                                                       GIS07350
       G=H+1                                                                     GIS07360
       NTACTY(G)=1.0                                                             GIS07370
       RAYY(G)=RAYY(H)                                                           GIS07380
       RAYZ(G)=RAYZ(H)+GRID                                                      GIS07390
       IF(FLAG(H) .EQ. 0) COUNT=COUNT+1                                          GIS07400
       GO TO 30                                                                  GIS07410
C                                                                                GIS07420
C      END OF RAY TRACE                                                          GIS07430
C                                                                                GIS07440
C      FLAG ALL FAILED RAYS                                                      GIS07450
C                                                                                GIS07460
C      NO INTERCEPT WITH 2ND SURFACE                                             GIS07470
 50    NTACTY(G)=0.0                                                             GIS07480
       FLAG(G)=1                                                                 GIS07490
       IF(SQURAY .EQ. 1) WRITE(6,600)G,RAYY(G),RAYZ(G)                           GIS07500
       GO TO 49                                                                  GIS07510
C      INTERCEPT OUTSIDE BOUNDARY OF SECOND SURFACE                              GIS07520
 51    NTACTY(G)=0.0                                                             GIS07530
       FLAG(G)=2                                                                 GIS07540
       IF(SQURAY .EQ. 1) WRITE(6,700)G,RAYY(G),RAYZ(G)                           GIS07550
       GO TO 49                                                                  GIS07560
C      TOTAL INTERNAL RELECTION AT 2ND SURFACE                                   GIS07570
 52    NTACTY(G)=0.0                                                             GIS07580
       FLAG(G)=3                                                                 GIS07590
       IF(SQURAY .EQ. 1) WRITE(6,800)G,RAYY(G),RAYZ(G)                           GIS07600
       GO TO 49                                                                  GIS07610
C      TOTAL EXTERNAL REFLECTION                                                 GIS07620
 53    NTACTY(G)=0.0                                                             GIS07630
       FLAG(G)=4                                                                 GIS07640
       IF(SQURAY .EQ. 1) WRITE(6,802) G,RAYY(G),RAYZ(G)                          GIS07650
       GO TO 49                                                                  GIS07660
C      RAY OUTSIDE LENS DIAMETER                                                 GIS07670
 55    NTACTY(G)=0.0                                                             GIS07680
```

```
      COUNT2=COUNT2+1                                                    GIS07690
      FLAG(G)=5                                                          GIS07700
      IF(SQURAY .EQ. 1) WRITE(6,804) G,RAYY(G),RAYZ(G)                   GIS07710
      GO TO 49                                                           GIS07720
C     FAILED INTER2 SUBROUTINE                                           GIS07730
 56   NTNCTY(G)=0.0                                                      GIS07740
      IF(SQURAY .EQ. 1) WRITE(6,806) G,RAYY(G),RAYZ(G)                   GIS07750
      GO TO 49                                                           GIS07760
C                                                                        GIS07770
C     MIRROR IMAGE RAY MATCHING FOLLOWS                                  GIS07780
C                                                                        GIS07790
   60 CONTINUE                                                           GIS07800
      IF(SQURAY .EQ.1) WRITE(6,401)                                      GIS07810
      Q=G-1                                                              GIS07820
      S=G-Q                                                              GIS07830
   61 CONTINUE                                                           GIS07840
      IF(RAYZ(S) .EQ. 0.0) S=S+1                                         GIS07850
        IF(S .GT. Q) GO TO 70                                            GIS07860
          FLAG(G)=FLAG(S)                                                GIS07870
          IF(FLAG(G) .EQ. 0) COUNT=COUNT+1                               GIS07880
          RAYY(G)=RAYY(S)                                                GIS07890
          RAYZ(G)=-RAYZ(S)                                               GIS07900
          IF(FLAG(S) .GE. 1) GO TO 63                                    GIS07910
            XDIAPT(G)=XDIAPT(S)                                          GIS07920
            YDIAPT(G)=YDIAPT(S)                                          GIS07930
            YIM(G)=YIM(S)                                                GIS07940
            ZIM(G)=-ZIM(S)                                               GIS07950
            OPL(G)=OPL(S)                                                GIS07960
            NTNCTY(G)=NTNCTY(S)                                          GIS07970
            IF(SQURAY .EQ. 1) WRITE(6,501)G,S,RAYY(G),RAYZ(G),OPL(G),    GIS07980
     1      YIM(G),ZIM(G),NTNCTY(G)                                      GIS07990
   62 CONTINUE                                                           GIS08000
      S=S+1                                                              GIS08010
      G=G+1                                                              GIS08020
      GO TO 61                                                           GIS08030
C                                                                        GIS08040
C    FLAG ALL MIRROR IMAGE RAYS                                          GIS08050
C                                                                        GIS08060
C     NO INTERCEPT WITH 2ND SURFACE                                      GIS08070
   63 IF(FLAG(S) .GT. 1) GO TO 64                                        GIS08080
        IF(SOLRAY .EQ. 1 ) WRITE(6,601)G,S,RAYY(G),RAYZ(G)               GIS08090
        GO TO 62                                                         GIS08100
C     INTERCEPT OUTSIDE BOUNDARY OF 2ND SURFACE                          GIS08110
   64 IF(FLAG(S) .GT. 2) GO TO 65                                        GIS08120
        IF(SOLRAY .EQ. 1) WRITE(6,701)G,S,RAYY(G),RAYZ(G)                GIS08130
        GO TO 62                                                         GIS08140
C     TOTAL EXTERNAL REFLECTION                                          GIS08150
   65 IF(FLAG(S) .GT. 3) GO TO 66                                        GIS08160
```

```
      IF(SQURAY .EQ. 1) WRITE(6,801)G,S,RAYY(G),RAYZ(G)                 GISO8170
      GO TO 62                                                          GISO8180
C     CONTINUE                                                          GISO8190
   66 IF(FLAG(S) .GT. 4) GO TO 67                                       GISO8200
      IF(SQURAY .EQ. 1) WRITE(6,803) G,S,RAYY(G),RAYZ(G)                GISO8210
      GO TO 62                                                          GISO8220
C     RAY OUTSIDE LENS DIAMETER                                         GISO8230
   67 IF(SQURAY .EQ. 1) WRITE(6,805) G,S,RAYY(G),RAYZ(G)                GISO8240
      COUNT2=COUNT2+1                                                   GISO8250
      GO TO 62                                                          GISO8260
   70 CONTINUE                                                          GISO8270
      RAYS=G-1                                                          GISO8280
      DRAYS=RAYS-C                                                      GISO8290
      WRITE(6,1350) OSYMB,RZERO                                         GISO8300
      WRITE(6,500)RAYS,DRAYS,COUNT2,COUNT                               GISO8310
      XI=1.0                                                            GISO8320
      YI=1.0                                                            GISO8330
      ZI=1.0                                                            GISO8340
      WRITE(4,127) XI,YI,ZI,CKPP,CLPP,CMPP                              GISO8350
  127 FORMAT(' ',6F10.7)                                                GISO8360
      WRITE(4,128)                                                      GISO8370
  128 FORMAT(' ','/*')                                                  GISO8380
C                                                                       GISO8390
C     GENERATE OBJECT PLANE ELLIPSES:                                   GISO8400
C                                                                       GISO8410
      IF(ELLIPS .EQ. 1) WRITE(6,1000)ENUM                               GISO8420
      NUMBR=1                                                           GISO8430
      NUME=1                                                            GISO8440
   71 CONTINUE                                                          GISO8450
      COORD=1                                                           GISO8460
      ELZ(NUME,COORD)=-Y1(NUMBR)                                        GISO8470
   72 CONTINUE                                                          GISO8480
      ELY(NUME,COORD)=(X1(NUMBR)+AB)*SIN(ALFAP)+DSQRT((Y1(NUMBR))**2-   GISO8490
     1(ELZ(NUMB,COORD))**2)*COS(ALFAP)                                  GISO8500
      ELZZ=ELZ(NUMB,COORD)                                              GISO8510
      IF(ELLIPS .EQ. 1) WRITE(6,1100)NUMBR,COORD,ELY(NUMB,COORD),       GISO8520
     1ELZ(NUME,COORD)                                                   GISO8530
      COORD=COORD+1                                                     GISO8540
      ELZ(NUME,COORD)=ELZZ+ELL                                          GISO8550
      IF(ELZ(NUMB,COORD) .GE. Y1(NUMBR)) GO TO 73                       GISO8560
        GO TO 72                                                        GISO8570
   73 CONTINUE                                                          GISO8580
      ELZZ=ELZ(NUMB,COORD)                                              GISO8590
   74 CONTINUE                                                          GISO8600
      ELZ(NUME,COORD)=ELZZ-ELL                                          GISO8610
      IF(ELZ(NUMB,COORD) .LE. -Y1(NUMBR)) GO TO 75                      GISO8620
      ELY(NUME,COORD)=(X1(NUMBR)+AB)*SIN(ALFAP)-DSQRT((Y1(NUMBR))**2-   GISO8630
     1(ELZ(NUME,COORD))**2)*COS(ALFAP)                                  GISO8640
```

```
      ELZZ=ELZ(NUMB,COORD)                                                  GISO8650
      IF(ELLIPS .EQ. 1) WRITE(6,1100)NUMBR,COORD,ELY(NUMB,COORD),           GISO8660
     1ELZ(NUMB,COORD)                                                       GISO8670
      COOR(NUMB)=COORD                                                      GISO8680
      COORD=COORD+1                                                         GISO8690
      GO TO 74                                                              GISO8700
   75 CONTINUE                                                              GISO8710
      NUMBR=NUMBR+(K-1)/ELLNUM                                              GISO8720
      NUMB=NUMB+1                                                           GISO8730
      IF(NUMBR .GT. K) GO TO 76                                             GISO8740
      GO TO 71                                                              GISO8750
   76 CONTINUE                                                              GISO8760
      IF(ELLIPS .EQ. 1) WRITE(6,1200)                                       GISO8770
      NUMBT=NUMB-1                                                          GISO8780
C                                                                           GISO8790
C     IMAGE PLANE SPOT DIAGRAM STATISTICAL ANALYSIS:                        GISO8800
C                                                                           GISO8810
      SUM1=0.0                                                              GISO8820
      DO 77 G=1,RAYS                                                        GISO8830
        IF(FLAG(G) .GT. 0) GO TO 77                                         GISO8840
          SUM1=YIM(G)+SUM1                                                  GISO8850
   77 CONTINUE                                                              GISO8860
C     REFERENCE EACH ITERATION TO INCIDENT ANGLE & THICKNESS BY:            GISO8870
      IANGLE=ALFAP*10+1                                                     GISO8880
      THICK=(T/R)*100                                                       GISO8890
C     CENTROID OF SPOT:                                                     GISO8900
      YCENTR(IANGLE,THICK)=SUM1/COUNT                                       GISO8910
C     STANDARD DEVIATIONS:                                                  GISO8920
      SUM2=0.0                                                              GISO8930
      SUM3=0.0                                                              GISO8940
      DO 78 G=1,RAYS                                                        GISO8950
        IF(FLAG(G) .GT. 0) GO TO 78                                         GISO8960
        SUM2=ZIM(G)**2+SUM2                                                 GISO8970
        SUM3=(YCENTR(IANGLE,THICK)-YIM(G))**2+SUM3                          GISO8980
   78 CONTINUE                                                              GISO8990
      SIGMAZ(IANGLE,THICK)=SUM2/COUNT                                       GISO9000
      SIGMAY(IANGLE,THICK)=SUM3/COUNT                                       GISO9010
C     ROOT MEAN SQUARE SPOT SIZE:                                           GISO9020
      RMSRAD(IANGLE,THICK)=SQRT(SIGMAZ(IANGLE,THICK)+SIGMAY(IANGLE,         GISO9030
     1THICK))                                                               GISO9040
      WRITE(6,1300) T,U,ALFAP,R,YCENTR(IANGLE,THICK),                       GISO9050
     1SIGMAY(IANGLE,THICK),SIGMAZ(IANGLE,THICK),RMSRAD(IANGLE,THICK)        GISO9060
C                                                                           GISO9070
C     SPOT DIAGRAM ENERGY DENSITY VS. RADIUS FROM CENTROID:                 GISO9080
C                                                                           GISO9090
      DO 79 G=1,RAYS                                                        GISO9100
        IF(FLAG(G) .GT. 0) GO TO 79                                         GISO9110
        SDRAD(G)=DSQRT((YCENTR(IANGLE,THICK)-YIM(G))**2+ZIM(G)**2)          GISO9120
```

```
   79 CONTINUE                                                          GISO9130
      GG=1                                                              GISO9140
      ROC(GG)=0.005                                                     GISO9150
      SUM4(GG)=0.0                                                      GISO9160
   80 CONTINUE                                                          GISO9170
      DO 81 G=1,RAYS                                                    GISO9180
        IF(FLAG(G) .GT. 0) GO TO 81                                     GISO9190
        IF(SRAD(G) .LE. ROC(GG)) SUM4(GG)=1+SUM4(GG)                    GISO9200
   81 CONTINUE                                                          GISO9210
      FRACTN(GG)=SUM4(GG)/RAYS                                          GISO9220
C     WRITE(6,1400) GG,ROC(GG),FRACTN(GG)                               GISO9230
      HH=GG                                                             GISO9240
      GG=GG+1                                                           GISO9250
      ROC(GG)=ROC(HH)+0.005                                             GISO9260
      SUM4(GG)=0.0                                                      GISO9270
      IF(ROC(GG) .GT. 0.5) GO TO 82                                     GISO9280
      GO TO 80                                                          GISO9290
   82 CONTINUE                                                          GISO9300
      GGT=GG-1                                                          GISO9310
      WRITE(6,1500)                                                     GISO9320
      WRITE(6,1600)                                                     GISO9330
C                                                                       GISO9340
C*********************************************************************  GISO9350
C  END OF PROGRAM                                                       GISO9360
C*********************************************************************  GISO9370
C                                                                       GISO9380
C     WRITE OUTPUT FOR PLOTTING PURPOSES                                GISO9390
C                                                                       GISO9400
C                                                                       GISO9410
      WRITE(1,83)F,ALPHA,U,I,R,T,N1,N2,N3,ALFAP,GRID,STATNA,GAMMA,RAYS, GISO9420
     1COUNT,K,GGT,IANGLE,THICK,OSYMB,TSURFL,BEFF,PRCNTO,PRCNTE,PRCNTT,  GISO9430
     1XMAVE,A,E                                                         GISO9440
   83 FORMAT(' ',3F9.7,I4,4F9.7/' ',3F9.7,F9.5,F9.7/' ',6I4,4F11.7/1X,  GISO9450
     15F11.7)                                                           GISO9460
      DO 85 J=1,K                                                       GISO9470
         WRITE(1,84)X1(J),Y1(J),X2(J),Y2(J)                             GISO9480
   84    FORMAT(' ',4F10.7)                                            GISO9490
   85 CONTINUE                                                          GISO9500
      DO 88 G=1,RAYS                                                    GISO9510
         WRITE(1,86)RAYY(G),RAYZ(G),FLAG(G)                             GISO9520
   86    FORMAT(' ',2F10.7,3X,I1)                                      GISO9530
         IF(FLAG(G) .GT. 0) GO TO 88                                    GISO9540
           WRITE(1,87)YIM(G),ZIM(G)                                     GISO9550
   87      FORMAT(' ',2F10.7)                                           GISO9560
   88 CONTINUE                                                          GISO9570
      DO 90 GG=1,GGT                                                    GISO9580
         WRITE(1,89)ROC(GG),FRACTN(GG)                                  GISO9590
   89    FORMAT(' ',F10.7,2X,F10.7)                                    GISO9600
```

```
   90 CONTINUE                                                          GIS09610
      WRITE(1,91)YCENTR(IANGLE,THICK),SIGMAY(IANGLE,THICK),             GIS09620
     1SIGMAZ(IANGLE,THICK),RMSRAD(IANGLE,THICK),NUMBT                   GIS09630
   91 FORMAT(' ',4F10.7,I4)                                             GIS09640
      DO 96 NUMB=1,NUMBT                                                GIS09650
         WRITE(1,92)COOR(NUMB)                                          GIS09660
   92    FORMAT(' ',I3)                                                GIS09670
         COORT=COOR(NUMB)                                               GIS09680
         DO 95 COORD=1,COORT                                            GIS09690
            WRITE(1,94)ELY(NUMB,COORD),ELZ(NUMB,COORD)                  GIS09700
   94       FORMAT(' ',F11.8,3X,F11.8)                                  GIS09710
   95    CONTINUE                                                       GIS09720
   96 CONTINUE                                                          GIS09730
      WRITE(1,97)                                                       GIS09740
   97 FORMAT(' ','/*')                                                  GIS09750
C                                                                       GIS09760
C    FORMAT STATMENTS                                                   GIS09770
C                                                                       GIS09780
  10C FORMAT('1',' LENS PARAMETERS:'///5X,'ALPHA = ',F9.7,2X,           GIS09790
     1'BETA = ',F9.7,2X,'RADIUS = ',F7.5//5X,'INCIDENT ANGLE = ',       GIS09800
     1F9.6,2X,'ITERATIONS = ',I5//5X,'EDGE THICKNESS = ',F9.7,2X,       GIS09810
     1'INDICES OF REFRACTION: N1 = ',F7.5//5X,'N3 = ',                  GIS09820
     1F7.5,2X,'FOCAL LENGTH FROM STATION ZERO = ',F7.5,2X,              GIS09830
     1'DELTA UDP = ',F7.5///                                            GIS09840
     11X,'  J        X1          Y1          X2          Y2          UDP         I2PGIS09850
     1        I2'//25X,'LP          I1P          I1      DYDXT'//)     GIS09860
 200 FORMAT(1X,I4,7F10.7/20X,4F10.7/)                                   GIS09870
 30C FORMAT('C'/1X,'NOSE SECTION DATA:'//2X,'NOSE HALF ANGLE = ',       GIS09880
     1F7.5,2X,'STATION A = ',F9.5/2X,'OPAQUE SURFACE LENGTH = ',F7.5/2X,GIS09890
     1'TRANSPARENT SURFACE LENGTH =',F7.5/2X,'TOTAL LENS SURFACE LENGTH GIS09900
     1=',F7.5///)                                                       GIS09910
 400 FORMAT('1',' SKEW RAY TRACE PARAMETERS:'///5X,'ALFAP= ',F9.7,2X,   GIS09920
     1'GRID= ',F9.7,2X,' SEE LENS PARAMETERS ABOVE.'///2X,'RAY     RAYY  GIS09930
     1    RAYZ          XO          YO          XI          YI         ZI'//14X,'OGIS09940
     1PL          YIM          ZIM      NTNCTY       XDIAPT       YDIAPT'///)GIS09950
 401 FORMAT('1'/////'MIRROR IMAGE SKEW RAYS FOLLOW'///1X,'IMAGE'/2X,'RAGIS09960
     1Y   RAY    RAYY        RAYZ          CPL          YIM          ZIM      NTNCTGIS09970
     1Y'//)                                                             GIS09980
 50C FORMAT(1X,I4,7F10.7/8X,F13.9,3F10.7,2E13.3/)                       GIS09990
 501 FORMAT(I4,1X,I4,2F10.7,F13.9,3F10.7/)                              GIS10000
 60C FORMAT(1X,I4,2F10.7,' NO INTERCEPT WITH 2ND SURFACE'/)             GIS10010
 601 FORMAT(I4,1X,I4,2F10.7,' NO INTERCEPT WITH 2ND SURFACE'/)          GIS10020
 70C FORMAT(1X,I4,2F10.7,' INTERCEPT OUTSIDE BOUNDARY OF 2ND SURFACE'/)GIS10030
 701 FORMAT(I4,1X,I4,2F10.7,' INTERCEPT OUTSIDE OF BOUNDARY OF 2ND SURFGIS10040
     1ACE'/)                                                            GIS10050
 800 FORMAT(1X,I4,2F10.7,2X,'TOTAL INTERNAL REFLECTION AT 2ND SURFACE'/GIS10060
     1)                                                                 GIS10070
 801 FORMAT(I4,1X,I4,2F10.7,'  TOTAL INTERNAL REFLECTION'/)             GIS10080
```

```
  802 FORMAT(1X,I4,2F10.7,2X,'  TOTAL EXTERNAL REFLECTION'/)                GIS1C090
  803 FORMAT(I4,1X,I4,2F10.7,'  TOTAL EXTERNAL REFLECTION'/)                GIS10100
  804 FORMAT(1X,I4,2F10.7,2X,'  RAY OUTSIDE LENS DIAMETER'/)                GIS1C110
  805 FORMAT(I4,1X,I4,2F10.7,'  RAY OUTSIDE LENS DIAMETER'/)                GIS10120
  806 FORMAT(1X,I4,2F10.7,' FAILED RAY IN INTER2 ARSIN'/)                   GIS1C130
  900 FORMAT(1X,'END OF SKEW RAY TRACE.'//3X,'TOTAL NUMBER OF RAYS TRACEGIS1C140
     1D = ',I4/3X,'TOTAL NUMBER OF MIRROR IMAGE RAYS = ',I4/3X,'TOTAL NUGIS10150
     1MBER OF RAYS OUTSIDE LENS DIAMETER = 'I4/3X,'TOTAL NUMBER OF RAYS GIS1C160
     1STRIKING IMAGE PLANE = ',I4)                                        GIS10170
 1000 FORMAT('1','TWO DIMENSIONAL PROJECTION OF 3-DIM LENS'///3X,'GENERAGIS1C180
     1TED ELLIPSES ARE REFERENCED TO THE CORRESPONDING JTH COORD'/3X,'INGIS10190
     1THE MERIDIAN PLANE AS COMPUTED IN THE ABOVE'/3X,'GLM LENS SHAPEAL GIS10200
     1GORITHM. INSIDE/OUTSIDE ELLIPSES DEFINE THE '/3X,'LIMITS OF THE TRGIS1C210
     1ANSPARENT/OPAQUE PARTS.'/3X,'THE NUMBER OF ELLIPSES =  ',I3//1X,'  GIS10220
     1 J    COORD     ELY          ELZ'//)                                 GIS1C230
 1100 FORMAT(1X,I4,3X,I4,2F10.7)                                          GIS1C240
 1200 FORMAT(1X/'CALCULATION OF ELLIPSE COORDINATES COMPLETE.')            GIS10250
 1300 FORMAT(1X,'IMAGE PLANE SPOT DIAGRAM ANALYSIS:'/1X,'THICKNESS      GIS1C260
     1 =',F9.7,'   U =',F9.7,'   ALFAP =',F9.7,'   R =',F9.7/1X,'CENTROID:  GIS1C270
     1 ZCENTR = 0.0,   YCENTR = ',F10.7/1X,'STANDARD DEVIATIONS:  SIGMAY GIS10280
     1 = ',F10.7,2X,'SIGMAZ = ',F10.7/1X,'RMS SPOT SIZE: RMSRAD = ',     GIS1C290
     1F10.7)                                                              GIS10300
 1350 FORMAT(1X,//1X,'CSYMB =',1X,F6.3,/1X,'RZERO=',1X,F6.3)              GIS1C310
 1400 FORMAT(I4,2F11.7)                                                   GIS10320
 1500 FORMAT(1X/'SPOT DIAGRAM CALCULATIONS COMPLETE.'////)                GIS10330
 1600 FORMAT(1X/'END OF PROGRAM'////)                                     GIS1C340
 1700 FORMAT(1X,'LENS INDEX OF REFRACTION DATA'/2X,'THE INDEX GRADIENT IGIS10350
     1S: N2(R) = SQRT((',F4.2,')+(',F6.3,')*R**2)'/2X,'WITH THE CENTER OGIS10360
     1F SYMMETRY LOCATED AT ',F7.3/2X,'(MINUS IMPLIES THE OSYM IS ON THEGIS1C370
     1 IMAGE SIDE).'/2X,'PERCENT CHANGE OF THE LENS INDEX GRADIENT:'/    GIS10380
     14X,'INSIDE SURFACE FROM CENTER TO EDGE :',F10.5/4X,'INSIDE SURFACEGIS1C390
     1 TO OUTSIDE SURFACE AT THICKEST POINT :',F10.5/4X,'OUTSIDE SURFACEGIS10400
     1 FROM CENTER TO EDGE :',F10.5////)                                  GIS10410
 1800 FORMAT(1X,'IMAGINARY RADIUS DUE TO IMPROPER A TO B RATIO IN N2(R) GIS1C420
     1FOR THIS RAY')                                                      GIS10430
 1900 FORMAT(1X,'**************************************************GIS10440
     1**************'/1X,'*',68X,'*'/1X,'*',25X,'GISL TABULAR OUTPUT',2GIS1C450
     14X,'*'/1X,'*',68X,'*'/1X,'**************************************GIS10460
     1*****************************'///)                                  GIS1C470
      STOP                                                                GIS10480
      END                                                                 GIS10490
C                                                                         GIS1C500
C                                                                         GIS1C510
C     SUBROUTINE INDEX CALCULATES THE INDEX OF REFRACTION AS A FUNCTION   GIS1C520
C      OF RADIUS FROM THE CENTER OF SYMMETRY. THE VALUES OF THE INDEX     GIS1C530
C      CONSTANTS A AND B ARE USER INPUTS.                                 GIS10540
C                                                                         GIS1C550
      SUBROUTINE INDEX(R,N,A,B,RZERO)                                     GIS1C560
```

```
      DOUBLE PRECISION R,N,A,B,RZERO                                    GIS10570
      N=DSQRT(A+B*(R/RZERO)**2)                                         GIS10580
      RETURN                                                            GIS10590
      END                                                               GIS10600
C                                                                       GIS10610
C                                                                       GIS10620
C     SUBROUTINE TRANSMIT CALCULATES THE TRANSMITTED INTENSITY OF       GIS10630
C     EACH RAY AT BOTH THE OUTSIDE AND INSIDE SURFACES USING THE        GIS10640
C     FRESNEL EQUATIONS.                                                GIS10650
C                                                                       GIS10660
      SUBROUTINE XMIT(PHI,N2,XM1,FACE,XM,N1,N3)                         GIS10670
      REAL PHI,N3,XM1,NTI,N1,N2,SQRT,COS                                GIS10680
      INTEGER FACE                                                      GIS10690
      NTI=N2/N1                                                         GIS10700
      IF(FACE .EQ. 2) NTI=N3/N2                                         GIS10710
      VALUE=NTI**2-(SIN(PHI)**2)                                        GIS10720
      IF(VALUE .LT. 0.0) GO TO 10                                       GIS10730
      SQ=SQRT(VALUE)                                                    GIS10740
      COSP=COS(PHI)                                                     GIS10750
      RPER=(COSP-SQ)/(COSP+SQ)                                          GIS10760
      RPAR=((NTI**2)*COSP-SQ)/((NTI**2)*COSP+SQ)                        GIS10770
      XM1=1.0-0.5*(RPER**2+RPAR**2)                                     GIS10780
      RETURN                                                            GIS10790
   10 CONTINUE                                                          GIS10800
      FACE=3                                                            GIS10810
      RETURN                                                            GIS10820
      END                                                               GIS10830
C                                                                       GIS10840
C                                                                       GIS10850
C     SUBROUTINE RADYUS CALCULATES THE RADIAL DIMENSION FROM THE CENTER GIS10860
C     OF SYMMETRY TO THE RAY AT THE ANGLE SPECIFIED.                    GIS10870
C     RADYUS ALSO FINDS D(R)/D(THETA) FOR SUBR ITRATE.                  GIS10880
C                                                                       GIS10890
      SUBROUTINE RADYUS(A,E,ARG,RSINO,ANGLE,R,DR,FILTER,EPSLON)         GIS10900
      REAL EPSLON                                                       GIS10910
      DOUBLE PRECISION ANGLE,R,DR,ARG,RSINO,FRACTN,A,E                  GIS10920
      INTEGER J,FILTER                                                  GIS10930
      FRACTN=DSIN(-2.0*EPSLON*ANGLE+RSINO)                              GIS10940
      IF((A+ARG*FRACTN) .LE. 0.0) GO TO 10                              GIS10950
      R=SQRT(2.0)*DABS(E)/DSQRT(A+ARG*FRACTN)                           GIS10960
      DR=((EPSLON*R**3)/(2.0*E**2))*DCOS(-2.0*EPSLON*ANGLE+RSINO)*ARG   GIS10970
   10 CONTINUE                                                          GIS10980
      FILTER=0                                                          GIS10990
      IF((A+ARG*FRACTN) .LT. 0.0) FILTER=1                              GIS11000
      RETURN                                                            GIS11010
      END                                                               GIS11020
C                                                                       GIS11030
C    SUBROUTINE RADUS CALCULATES THE RADIUS TO THE FRONT SURFACE AT     GIS11040
```

```
C     THE SPECIFIED ANGLE AND ALSO FINDS ITS DERIVATIVE WITH RESPECT      GIS11050
C     TO ANGLE.                                                           GIS11060
C                                                                         GIS11070
      SUBROUTINE RADUS(PI,X1,Y1,I1,U,CSYMB,THETAP,RADD,RDG)               GIS11080
      DOUBLE PRECISION X1,Y1,I1,U,THETAP,RADD,RDG,PI,PI2,CSYMB            GIS11090
      PI2=PI/2.                                                           GIS11100
      S=DCOTAN(I1+U)                                                      GIS11110
      B=Y1-S*(X1+CSYMB)                                                   GIS11120
      RADD=B/(DSIN(THETAP)-S*DCOS(THETAP))                                GIS11130
      RDG=-B*(DCOS(THETAP)+S*DSIN(THETAP))/                               GIS11140
     1((DSIN(THETAP)-S*DCOS(THETAP))**2)                                  GIS11150
      RETURN                                                              GIS11160
      END                                                                 GIS11170
C                                                                         GIS11180
C     SUBROUTINE INTERCEPT1 PERFORMS ITERATION TO FIND R AND THETAP       GIS11190
C     OF THE INTERCEPT OF THE GRIN SKEW-RAY AND THE INSIDE CONICAL        GIS11200
C     SURFACE USING THE NEWTON-RAPHSON ITERATION PROCEDURE.               GIS11210
C                                                                         GIS11220
      SUBROUTINE INTER1(SIGN,THETA,R,THETAP,FILTER,A,E,ARG,RSINO,         GIS11230
     1EPSLON,C,TCRIT,NPOX,NPOY,NPOZ,PSIO,XO,YO,ZO,RO,CKKP,CLLP,          GIS11240
     1CMMP,ALPHA,CSYMB,FLUG)                                              GIS11250
      INTEGER LOOP,FILTER,FLUG                                            GIS11260
      REAL SIGN,EPSLON                                                    GIS11270
      DOUBLE PRECISION THETA,R,THETAP,A,E,ARG,RSINO,RO,CSYMB,             GIS11280
     1ALPHA,RADG,RADH,FX,FXDOT,RDOTG,RDOTS,XNEW,DIF,C,TCRIT,PSIO,         GIS11290
     1FACTOR,EXIT,THET,PI,XO,YO,ZO,NPOX,NPOY,NPOZ,CKKP,CLLP,CMMP         GIS11300
C                                                                         GIS11310
      LOOP=0                                                              GIS11320
      FILTER=0                                                            GIS11330
      PI=3.14159265358979 3                                               GIS11340
   10 CONTINUE                                                            GIS11350
      THET=THETA                                                          GIS11360
      IF(THETA.GT.TCRIT) THET=THETA-C                                     GIS11370
      IF(THETA.GT.TCRIT) EPSLON=+1.0                                      GIS11380
      CALL RADYUS(A,E,ARG,RSINO,THET,RADG,RDOTG,FILTER,EPSLON)            GIS11390
      IF(FILTER .EQ. 1)RETURN                                             GIS11400
      IF(FILTER .EQ. 2)RETURN                                             GIS11410
      CALL SURF(THETA,RADH,RDOTS,SIGN,FILTER,NPOX,NPOY,NPOZ,PSIO,XO,      GIS11420
     1YO,ZO,CSYMB,RO,CKKP,CLLP,CMMP,ALPHA)                                GIS11430
      IF(FILTER .EQ. 3)RETURN                                             GIS11440
      FX=RADG-RADH                                                        GIS11450
      FXDOT=RDOTG-RDOTS                                                   GIS11460
      FACTOR=1.7/(DABS(PI-PSIO))**.5                                      GIS11470
      IF(CSYMB.GT.0.0) FACTOR=1.3                                         GIS11480
      XNEW=THETA-(FX/FXDOT)/FACTOR                                        GIS11490
      DIF=DABS(THETA-XNEW)                                                GIS11500
      EXIT=0.000001*DABS(PI-PSIO)                                         GIS11510
      IF(CSYMB.GT.0.0) EXIT=0.0000001                                     GIS11520
```

```
      IF(DIF .GT. EXIT) GO TO 20                                        GIS11530
      THETAP=THETA                                                      GIS11540
      R=RADG                                                            GIS11550
      RETURN                                                            GIS11560
   20 CONTINUE                                                          GIS11570
      IF(LOOP .GE. 90) GO TO 30                                         GIS11580
      LOOP=LOOP+1                                                       GIS11590
      THETA=XNEW                                                        GIS11600
      GO TO 10                                                          GIS11610
   30 CONTINUE                                                          GIS11620
      IF(LOOP .GE. 90) FILTER=3                                         GIS11630
      WRITE(6,6410)                                                     GIS11640
 6410 FORMAT(/1X,'SOLUTION NOT FOUND')                                  GIS11650
      RETURN                                                            GIS11660
      END                                                               GIS11670
C                                                                       GIS11680
C     SUBROUTINE INTERCEPT2 PERFORMS ITERATION TO FIND THE INTERSECTION  GIS11690
C     OF THE GRIN RAY AND THE INSIDE SURFACE USING THE ANGLE IN         GIS11700
C     TERMS OF RADIUS PROCEDURE.                                        GIS11710
C                                                                       GIS11720
      SUBROUTINE INTER2(RP,THETAP,RPP,THATP,A,E,ARG,RSINO,EPSLON,       GIS11730
     1SIGN,NPOX,NPOY,NPOZ,PSIO,XO,YO,ZO,CSYMB,RO,CKKP,CLLP,CMMP,        GIS11740
     1ALPHA,THETA,FILTER,FLUG)                                          GIS11750
      INTEGER TRIP,FILTER,FLUG                                          GIS11760
      REAL SIGN,EPSLON                                                  GIS11770
      DOUBLE PRECISION RP,THETAP,RPP,A,E,ARG,RSINO,OSYMB,               GIS11780
     1PSIO,RO,ALPHA,RSIN,CHECK1,CHECK2,THETA,THATP,RDOT,ARGH,           GIS11790
     1XO,YO,ZO,NPOX,NPOY,NPOZ,CKKP,CLLP,CMMP                            GIS11800
      FILTER=0                                                          GIS11810
      TRIP=0                                                            GIS11820
   10 TRIP=TRIP+1                                                       GIS11830
      ARGH=(2*E**2/RP**2-A)/ARG                                         GIS11840
      IF(DABS(ARGH).GT.1.0) GO TO 50                                    GIS11850
      RSIN=DARSIN(ARGH)                                                 GIS11860
      THETPP=-EPSLON/2*(RSIN-RSINO)                                     GIS11870
      CHECK1=THETPP-THETAP                                              GIS11880
      CALL SURF(THETPP,RPP,RDOT,SIGN,FILTER,NPOX,NPOY,NPOZ,PSIO,XO,     GIS11890
     1YO,ZO,CSYMB,RO,CKKP,CLLP,CMMP,ALPHA)                              GIS11900
      CHECK2=DABS(RP-RPP)                                               GIS11910
      RP=RPP                                                            GIS11920
      THETAP=THETPP                                                     GIS11930
      IF(TRIP.GT.100) GO TO 20                                          GIS11940
      IF(CHECK2.LT.0.000001) GO TO 20                                   GIS11950
      GO TO 10                                                          GIS11960
   20 THATP=THETAP                                                      GIS11970
   50 RETURN                                                            GIS11980
      END                                                               GIS11990
C                                                                       GIS12000
```

```
C      SUBROUTINE SURF CALCULATES BOTH THE RADIUS TO THE                    GIS12C10
CC    LCCUS OF THE INTERCEPT OF THE RAY PLANE AND THE IN-                   GIS12020
CC    SIDE SURFACE AND THE DERIVATIVE OF THE RADIUS WRT THETA               GIS12C30
CC    GIVEN THE ANGLE THETA. SURF IS DESIGNED PRIMARILY                     GIS12040
CC    FOR USE WITH SUBROUTINES INTERCEPT1 AND INTERCEPT2.                   GIS12050
C                                                                           GIS12060
      SUBROUTINE SURF(THETA,R,RDOT,SIGN,FILTER,NPCX,NPOY,NPOZ,PSIO,XO,      GIS12070
     1YO,ZO,CSYMB,RO,CKKP,CLLP,CMMP,ALPHA)                                  GIS12C80
      INTEGER FILTER                                                        GIS12C90
      REAL XC,YO,ZO,NPOX,NPOY,NPOZ,CKKP,CLLP,CMMP,SIGN                      GIS12100
      DOUBLE PRECISION THETA,R,RDOT,PSIO,CSYME,                             GIS12110
     1RO,ALPHA,ALFA,BETA,A,B,C,A2,B2,C2,SQUARE,                             GIS12120
     1DAL,DBE,DA,DB,DC,DA2,DB2,A3,B3,C3                                     GIS12130
      ALFA=DCOS(THETA)-DSIN(THETA)*DCOTAN(PSIO)                             GIS12140
      BETA=DSIN(THETA)/DSIN(PSIO)                                           GIS12150
      A=ALFA*(XC+CSYMB)/RO+BETA*CKKP                                        GIS12160
      B=ALFA*YC/RO+BETA*CLLP                                                GIS12170
      C=ALFA*ZC/RO+BETA*CMMP                                                GIS12180
      A2=B**2+C**2-(A**2)*(DTAN(ALPHA)**2)                                  GIS12190
      B2=B*NPOY+C*NPOZ+A*NPOX+2.0*A*CSYMB*(DTAN(ALPHA)**2)                  GIS12200
      C2=CSYME*NPOX+(CSYME*DTAN(ALPHA))**2+NPOX*XC+NPOY*YO+NPOZ*ZO          GIS12210
      SQUARE=0.25*(B2/A2)**2+C2/A2                                          GIS12220
      IF(SQUARE .LT. 0.0) FILTER=3                                          GIS12230
      IF(FILTER .EQ. 3) RETURN                                              GIS12240
      R=-B2/(2.0*A2)+SIGN*DSQRT(SQUARE)                                     GIS12250
C     CALCULATE DR/DTHETA                                                   GIS12260
      DAL=-DSIN(THETA)-DCOS(THETA)*DCOTAN(PSIO)                             GIS12270
      DBE=DCOS(THETA)/DSIN(PSIO)                                            GIS12280
      DA=DAL*(XC+CSYMB)/RO+DBE*CKKP                                         GIS12290
      DB=DAL*YC/RO+DBE*CLLP                                                 GIS12300
      DC=DAL*ZC/RO+DBE*CMMP                                                 GIS12310
      DA2=2.0*B*DB+2.0*C*DC-2.0*A*DA*(DTAN(ALPHA)**2)                       GIS12320
      DB2=DB*NPOY+DC*NPOZ+DA*NPOX+2.0*DA*CSYMB*(DTAN(ALPHA)**2)             GIS12330
      A3=-0.5*(A2*DB2-B2*DA2)/(A2**2)                                       GIS12340
      B3=SIGN/(2.0*DSQRT(0.25*(B2/A2)**2+C2/A2))                            GIS12350
      C3=0.25*((A2**2)*2.0*B2*DB2-(B2**2)*2.0*A2*DA2)/(A2**4)-              GIS12360
     1C2*DA2/(A2**2)                                                        GIS12370
      RDOT=A3+B3*C3                                                         GIS12380
      RETURN                                                                GIS12390
      END                                                                   GIS12400
C                                                                           GIS12410
CC                                                                          GIS12420
CC    SUBROUTINE DIRECT CALCULATES THE VECTOR DIRECTION COSINES OF          GIS12430
CC     A SKEW RAY AT THE POINT OF INTERCEPT WITH THE INSIDE SURFACE.        GIS12440
C                                                                           GIS12450
      SUBROUTINE DIRECT(CKP,CLP,CMP,DIR,RRX,RRY,RRZ,NPFX,NPFY,NPFZ,PSIR)   GIS12460
      DOUBLE PRECISION CKP,CLP,CMP,DIR,RRX,RRY,RRZ,NPFX,NPFY,NPFZ,PSIR,    GIS12470
     1EPS,APF,BPP,CPP,AA,BB,CC,AP,BP,CP                                     GIS12480
```

```
      EPS=1.0E-06                                                       GIS12490
      SIGN=+1.0                                                         GIS12500
      IF((DABS(NPFX)) .GT. EPS) GO TO 50                                GIS12510
      IF((DABS(NPFY)) .GT. EPS) GO TO 30                                GIS12520
      IF((DABS(RRY)) .LT. EPS) GO TO 25                                 GIS12530
        CMP=0.0                                                         GIS12540
      APP=1.0+(RRX/RRY)**2                                              GIS12550
      BPP=DCOS(PSIR)*RRX/RRY**2                                         GIS12560
      CPP=(DCOS(PSIR)/RRY)**2-1.0                                       GIS12570
        CKP=BPP/APP+SIGN*DSQRT(BPP**2-APP*CPP)/APP                      GIS12580
        CLP=(DCOS(PSIR)-RRX*CKP)/RRY                                    GIS12590
        RETURN                                                          GIS12600
   25 CONTINUE                                                          GIS12610
        CMP=0.0                                                         GIS12620
        CKP=DCOS(PSIR)/RRX                                              GIS12630
        CLP=SIGN*DSQRT(1.0-CKP**2)                                      GIS12640
        RETURN                                                          GIS12650
   30 CONTINUE                                                          GIS12660
      IF((DABS(NPFZ)) .GT. EPS) GO TO 40                                GIS12670
      IF((DABS(RRZ)) .LT. EPS) GO TO 35                                 GIS12680
        CLP=0.0                                                         GIS12690
      APP=1.0+(RRX/RRZ)**2                                              GIS12700
      BPP=DCOS(PSIR)*RRX/RRZ**2                                         GIS12710
      CPP=(DCOS(PSIR)/RRZ)**2-1.0                                       GIS12720
        CKP=BPP/APP+DSQRT(BPP**2-APP*CPP)/APP                           GIS12730
        CMP=(DCOS(PSIR)-RRX*CKP)/RRZ                                    GIS12740
        RETURN                                                          GIS12750
   35 CONTINUE                                                          GIS12760
        CLP=0.0                                                         GIS12770
        CKP=DCOS(PSIR)/RRX                                              GIS12780
        CMP=SIGN*DSQRT(1.0-CKP**2)                                      GIS12790
        RETURN                                                          GIS12800
   40 CONTINUE                                                          GIS12810
      IF((DABS(RRX)) .GT. EPS) GO TO 45                                 GIS12820
        CLP=DCOS(PSIR)/(RRY-NPFY/NPFZ*RRZ)                              GIS12830
        CMP=-NPFY/NPFZ*CLP                                              GIS12840
        CKP=SIGN*DSQRT(1.0-CLP**2-CMP**2)                               GIS12850
        RETURN                                                          GIS12860
   45 CONTINUE                                                          GIS12870
      IF((DABS(RRY-NPFY/NPFZ*RRZ)) .GT. EPS) GO TO 47                   GIS12880
        CKP=DCOS(PSIR)/RRX                                              GIS12890
        CLP=SIGN*DSQRT((1.0-CKP**2)/(1.0+(NPFY/NPFZ)**2))               GIS12900
        CMP=-NPFY/NPFZ*CLP                                              GIS12910
        RETURN                                                          GIS12920
   47 CONTINUE                                                          GIS12930
      SIGN=-1.0                                                         GIS12940
      IF(RRY .LT. 0.0 .OR. RRX .LT. 0.0) SIGN=+1.0                      GIS12950
      IF(RRY .LT. 0.0 .AND. RRX .LT. 0.0) SIGN=-1.0                     GIS12960
```

```
      APP=((RRY-NPFY/NPFZ*RRZ)/RRX)**2+1.0+(NPFY/NPFZ)**2                    GIS12970
      BPP=2.0*DCOS(PSIR)/(RRX**2)*(RRY-NPFY/NPFZ*RRZ)                        GIS12980
      CPP=(DCOS(PSIR)/RRX)**2-1.0                                            GIS12990
        CLP=EPP/(2.0*APP)+SIGN*DSQRT(BPP**2-4.0*APP*CPP)/(2.0*APP)           GIS13000
        CMP=-NPFY/NPFZ*CLP                                                   GIS13010
        CKP=(DCOS(PSIR)-(RRY-NPFY/NPFZ*RRZ)*CLP)/RRX                         GIS13020
        RETURN                                                               GIS13030
   5C CONTINUE                                                               GIS13040
      SIGN=-1.0                                                              GIS13050
      IF(RRY .LT. 0.0)SIGN=+1.0                                              GIS13060
      IF(RRY .LT. 0.0 .AND. NPFZ .LT. 0.0) SIGN=-1.0                         GIS13070
      AA=NPFX*RRY-NPFY*RRX                                                   GIS13080
      BB=NPFX*RRZ-NPFZ*RRX                                                   GIS13090
      CC=NPFX*DCOS(PSIR)                                                     GIS13100
      IF((DABS(BB)) .GT. EPS)GO TO 60                                        GIS13110
        CLP=CC/AA                                                            GIS13120
      APP=(NPFZ/NPFX)**2+1.0                                                 GIS13130
      BPP=NPFY*NPFZ/NPFX**2*CLP                                              GIS13140
      CPP=((NPFY/NPFX)**2+1.0)*CLP**2-1.0                                    GIS13150
        CMP=-EPP/APP+SIGN*DSQRT(BPP**2-APP*CPP)/APP                          GIS13160
        CKP=(-NPFY*CLP-NPFZ*CMP)/NPFX                                        GIS13170
        RETURN                                                               GIS13180
   6C CONTINUE                                                               GIS13190
      IF(BB .GT. 0.0) SIGN=-1.0                                              GIS13200
      AP=(NPFY/NPFX)**2+1.0                                                  GIS13210
      BP=(NPFZ/NPFX)**2+1.0                                                  GIS13220
      CP=2.0*NPFY*NPFZ/NPFX**2                                               GIS13230
      APP=AP+EP*(AA/BB)**2-CP*AA/BB                                          GIS13240
      BPP=2.0*AA*CC*BP/BB**2-CP*CC/BB                                        GIS13250
      CPP=BP*(CC/BB)**2-1.0                                                  GIS13260
        CLP=EPP/(2.0*APP)+SIGN*DSQRT(BPP**2-4.0*APP*CPP)/(2.0*APP)           GIS13270
        CMP=CC/BB-AA/BB*CLP                                                  GIS13280
        CKP=(-NPFY*CLP-NPFZ*CMP)/NPFX                                        GIS13290
        RETURN                                                               GIS13300
      END                                                                    GIS13310
```

```
C     ****************************************************************** CHE00C10
C     *                                                                * CHE00C20
C     *                        C H E C K                               * CHE00C30
C     *                                                                * CHE00C40
C     ****************************************************************** CHE00C50
C     *                                                                * CHE00C60
C     *        THIS PROGRAM IS DESIGNED TO CHECK THE ACCURACY OF       * CHE00C70
C     *    THE GISL FRONT SURFACE.   IT DOES THIS BY COMPLETING A      * CHE00080
C     *    RAY TRACE FROM THE MIDPOINT OF EACH SET OF POINTS THAT      * CHE00C90
C     *    DEFINE THE FRONT SURFACE, THROUGH THE LENS AND TOWARDS      * CHE00100
C     *    THE FOCAL POINT.   DEVIATIONS IN THE Y-COORDINATE AT        * CHE00110
C     *    THE FOCAL POINT ARE RECORDED.                               * CHE00120
C     ****************************************************************** CHE00130
C                                                                         CHE00140
C     MAJOR VARIABLE DEFINITIONS:                                       CHE00150
C                                                                       CHE00160
C        XO,YO     : COORDINATES OF MIDPOINT AT FRONT SURFACE          CHE00170
C        THETAO    : INITIAL ANGLE                                     CHE00180
C        RO        : INITIAL RADIUS                                    CHE00190
C        PSIO      : INITIAL ANGLE BETWEEN RADIUS AND RAY DIRECTION    CHE00200
C        C         : RAY MATCHING CONSTANT                             CHE00210
C        THETAC    : ANGLE OF MINIMUM RADIUS                           CHE00220
C        XI,YI     : FIRST GUESS COORDINATES AT INNER SURFACE          CHE00230
C        XF,YF     : SOLUTION POINT COORDINATES                        CHE00240
C        THETAP    : ANGLE AT INNER SURFACE                            CHE00250
C        RAD       : RADIUS AT INNER SURFACE                           CHE00260
C        PSIP      : ANGLE BETWEEN RADIUS AND RAY DIRECTION AT         CHE00270
C                    INNER SURFACE                                     CHE00280
C        YS        : Y-COORDINATE ON IMAGE PLANE                       CHE00290
C                                                                       CHE00300
C                                                                       CHE00310
C     SPECIFICATION STATEMENTS                                          CHE00320
C                                                                       CHE00330
      INTEGER K,I,L,TRIP,EXIT2                                          CHE00340
C                                                                       CHE00350
      DOUBLE PRECISION DIFF,THETAN,THETAP,THETAH,                       CHE00360
     1F,ALPHA,RL,N1,N2,N3,N2O,OSYMB,A,B,PI,PI2,BF,RZERO,                CHE00370
     1X1(1010),Y1(1010),X2(1010),Y2(1010),X2O,Y2O,RZ,XO,YO,XI,YI,       CHE00380
     1I1,I1P,I2,I2P,THETAO,RO,ARG,RSINO,C,TCRIT,PSIO,E,EPLSON,          CHE00390
     1S,S1,S2,S3,XF,YF,PSIP,ZETA,DP,YS(1010),SIN,EON,EONR,RP,RPP,BE,    CHE00400
     1SL,RSIN,THETA,CHECK1,CHECK2                                       CHE00410
C                                                                       CHE00420
C  READ INPUT FROM PROGRAM GISL                                         CHE00430
C                                                                       CHE00440
      READ(5,100)A,B,OSYMB,K,ALPHA,N3,F,RL                              CHE00450
      DO 5 J=1,K                                                        CHE00460
      READ(5,200)X1(J),Y1(J),X2(J),Y2(J)                                CHE00470
    5 CONTINUE                                                          CHE00480
```

```
C                                                                       CHE00490
C   SET CONSTANTS                                                       CHE00500
C                                                                       CHE00510
      PI=3.14159265358979                                               CHE00520
      PI2=PI/2                                                          CHE00530
      N1=1.0                                                            CHE00540
      N3=1.0                                                            CHE00550
      BF=F/RL                                                           CHE00560
      RZERO=((X2(1)+OSYMB)**2+Y2(1)**2)**.5                             CHE00570
      IF(OSYMB.LT.-0.6) RZERO=DSQRT(R**2+OSYMB**2)                      CHE00580
C                                                                       CHE00590
C   START RAY TRACE                                                     CHE00600
C                                                                       CHE00610
      DO 10 J=2,K                                                       CHE00620
      L=J-1                                                             CHE00630
      TRIP=0                                                            CHE00640
C                                                                       CHE00650
C   CALCULATE MIDPOINT                                                  CHE00660
C                                                                       CHE00670
      S=(Y1(J)-Y1(L))/(X1(J)-X1(L))                                     CHE00680
      XO=(X1(J)+X1(L))/2.                                               CHE00690
      YO=(Y1(J)+Y1(L))/2.                                               CHE00700
C                                                                       CHE00710
C   CALCULATE LINE CONSTANTS                                            CHE00720
C                                                                       CHE00730
      I1=PI2-DATAN(S)                                                   CHE00740
      THETAO=DATAN(YO/(XO+OSYMB))                                       CHE00750
      IF(THETAO.LT.0.0) THETAO=PI+THETAO                                CHE00760
      RO=DSQRT(YO**2+(XO+OSYMB)**2)                                     CHE00770
      CALL INDEX(RO,N2O,A,B,RZERO)                                      CHE00780
      I1P=DARSIN((N1/N2O)*DSIN(I1))                                     CHE00790
      PSIO=PI-THETAO-I1+I1P                                             CHE00800
      EPSLON=1.0                                                        CHE00810
      IF(PSIO.GT.PI2.OR.PSIO.LT.0.0) EPSLON=-1.0                        CHE00820
      E=RO*N2O*DSIN(PSIO)                                               CHE00830
      ARG=DSQRT(A**2+4*B*(E**2)/(RZERO**2))                             CHE00840
      RSINO=DARSIN((2*(E**2)/(RO**2)-A)/ARG)                            CHE00850
      C=PI2-RSINO                                                       CHE00860
      TCRIT=THETAO-(EPSLON/2.)*(PI2-RSINO)                              CHE00870
      IF(PSIO.LT.PI2) C=0.0                                             CHE00880
C                                                                       CHE00890
C   FIND FIRST GUESS COORDINATES, RADIUS AND ANGLE                      CHE00900
C                                                                       CHE00910
      S1=-DTAN(I1-I1P)                                                  CHE00920
      S2=DTAN(ALPHA)                                                    CHE00930
      XI=(YO-S1*XO)/(S2-S1)                                             CHE00940
      YI=S2*XI                                                          CHE00950
      THETAP=DATAN(YI/(XI+OSYMB))                                       CHE00960
```

```
      IF(THETAP.LT.0.0) THETAP=THETAP+PI                                CHE00970
      RP=DSQRT(YI**2+(XI+OSYMB)**2)                                     CHE00980
      EE=-OSYME                                                         CHE00990
      SL=1.0                                                            CHE01000
C                                                                       CHE01010
C     GO TO THETA IN TERMS OF RADIUS ITERATION IF ANGLE LARGE           CHE01020
C                                                                       CHE01030
      IF(THETAP.GT.2.0) GO TO 25                                        CHE01040
C                                                                       CHE01050
C     NEWTON-RAPHSON ITERATION                                          CHE01060
C                                                                       CHE01070
   20 CONTINUE                                                          CHE01080
         TRIP=TRIP+1                                                    CHE01090
         IF(THETAP.GE.TCRIT) THETP=THETAP-C                             CHE01100
         IF(THETAP.LT.TCRIT) THETP=THETAP                               CHE01110
         IF(THETAP.GE.TCRIT) EPSLON=+1.0                                CHE01120
         THETAH=THETP-THETA0                                            CHE01130
         CALL RADYOS(ARG,RSINO,THETAH,RAD,RDG,FILTER,A,EPSLON,E)        CHE01140
         CALL RADLS(OSYMB,S2,THETAP,RADD,RDGG)                          CHE01150
         RDF=RAD-RADD                                                   CHE01160
         RDT=RDG-RDGG                                                   CHE01170
         FACTOR=1.7/(DABS(PSIO))                                        CHE01180
         IF(FACTOR.GT.10.0) FACTOR=10.0                                 CHE01190
         IF(OSYMB.GT.0.0) FACTOR=1.3                                    CHE01200
         THETAN=THETAP-RDF/(RDT*FACTOR)                                 CHE01210
         DIFF=DABS(THETAN-THETAP)                                       CHE01220
         EXIT=0.00001*DABS(PSIO)                                        CHE01230
         IF(OSYMB.GT.0.0) EXIT=0.000001                                 CHE01240
         EXIT2=200                                                      CHE01250
         IF(DIFF.LT.EXIT) GO TO 30                                      CHE01260
         IF(TRIP.GT.EXIT2) GO TO 30                                     CHE01270
         THETAP=THETAN                                                  CHE01280
         GO TO 20                                                       CHE01290
C                                                                       CHE01300
C     THETA IN TERMS OF RADIUS ITERATION                                CHE01310
C                                                                       CHE01320
   25    TRIP=TRIP+1                                                    CHE01330
         RSIN=DARSIN((2*E**2/RP**2-A)/ARG)                              CHE01340
         THETA=THETA0-EPSLON/2*(RSIN-RSINO)                             CHE01350
         CHECK1=THETA-THETAP                                            CHE01360
         RPP=EE/(DSIN(THETA)-SL*DCOS(THETA))                            CHE01370
         CHECK2=DABS(RP-RPP)                                            CHE01380
         THETAP=THETA                                                   CHE01390
         RP=RPP                                                         CHE01400
         IF(TRIP.GT.100) GO TO 26                                       CHE01410
         IF(CHECK2.LT.0.0000001) GO TO 26                               CHE01420
         GO TO 25                                                       CHE01430
   26    RAD=RPP                                                        CHE01440
```

```
C                                                                       CHE01450
C   CALCULATE FINAL POINT AT INNER SURFACE COORDINATES AND ANGLES       CHE01460
C                                                                       CHE01470
   30 XF=RAD*DCOS(THETAP)-OSYMB                                         CHE01480
      YF=RAD*DSIN(THETAP)                                               CHE01490
      CALL INDEX(RAD,N2,A,B,RZERO)                                      CHE01500
      EONR=E/(N2*RAD)                                                   CHE01510
      IF(EONR.GT.1.0) WRITE(6,630)                                      CHE01520
      IF(EONR.GT.1.0) EONR=1.0                                          CHE01530
      PSIP=DARSIN(EONR)                                                 CHE01540
      IF(THETAP.LT.TCRIT.AND.PSIO.GT.0.0) PSIP=PI-PSIP                  CHE01550
      ZETA=PI-THETAP-PSIP                                               CHE01560
      I2=PI2-ZETA-ALPHA                                                 CHE01570
      EON=(N2/N3)*DSIN(I2)                                              CHE01580
      IF(EON.GT.1.0) WRITE(6,640)                                       CHE01590
      IF(EON.GT.1.0) EON=1.0                                            CHE01600
      I2P=DARSIN(EON)                                                   CHE01610
      UP=PI2-ALPHA-I2P                                                  CHE01620
C                                                                       CHE01630
C   CALCULATE Y-COORDINATE AT FOCAL POINT                               CHE01640
C                                                                       CHE01650
      S3=-DTAN(UP)                                                      CHE01660
      YS(J)=S3*(F-XF)+YF                                                CHE01670
C                                                                       CHE01680
C   WRITE OUTPUT                                                        CHE01690
C                                                                       CHE01700
      WRITE(6,500) J,PSIO,YS(J),TRIP                                    CHE01710
   10 CONTINUE                                                          CHE01720
      STOP                                                              CHE01730
C                                                                       CHE01740
C   FORMAT STATEMENTS                                                   CHE01750
C                                                                       CHE01760
  100 FORMAT(1X,3F11.7,I5,F10.7,3F5.3)                                  CHE01770
  200 FORMAT(1X,4F10.7)                                                 CHE01780
  500 FORMAT(1X,I6,2F16.8,I6)                                           CHE01790
  630 FORMAT(1X,'WHOOPS PSIP ARSIN')                                    CHE01800
  640 FORMAT(1X,'WHOOPS AGAIN I2P ARSIN')                               CHE01810
C                                                                       CHE01820
      END                                                               CHE01830
C                                                                       CHE01840
C     SUBROUTINE INDEX CALCULATES THE INDEX OF REFRACTION AS A FUNCTION CHE01850
C      OF RADIUS FROM THE CENTER OF SYMMETRY. THE VALUES OF THE INDEX   CHE01860
C      CONSTANTS A AND B ARE USER INPUTS.                               CHE01870
C                                                                       CHE01880
      SUBROUTINE INDEX(R,N,A,B,RZERO)                                   CHE01890
      DOUBLE PRECISION N,A,B,R,RZERO                                    CHE01900
      N=DSQRT(A+B*(R/RZERO)**2)                                         CHE01910
      RETURN                                                            CHE01920
```

```
      END                                                                CHE01930
C                                                                        CHE01940
C     SUBROUTINE RADYUS CALCULATES THE RADIUS FOR THE GRIN EQUATION AT   CHE01950
C      THE SPECIFIED ANGLE AS WELL AS ITS DERIVATIVE WITH RESPECT TO     CHE01960
C      ANGLE.                                                            CHE01970
C                                                                        CHE01980
      SUBROUTINE RADYUS(ARG,RSINO,ANGLE,R,DR,FILTER,A,EPSLON,E)          CHE01990
      INTEGER FILTER                                                     CHE02000
      DOUBLE PRECISION ANGLE,R,DR,ARG,RSINO,A,EPSLON,E,FRACTN            CHE02010
      FRACTN=DSIN(-2.0*EPSLON*ANGLE+RSINO)                               CHE02020
      IF((A+ARG*FRACTN) .LE. 0.0) GO TO 10                               CHE02030
      R=SQRT(2.0)*DABS(E)/DSQRT(A+ARG*FRACTN)                            CHE02040
      IF(FRACTN .LE. (-0.989)) GO TO 10                                  CHE02050
      DR=((EPSLON*R**3)/(2.0*E**2))*DCOS(-2.0*EPSLON*ANGLE+RSINO)        CHE02060
   10 CONTINUE                                                           CHE02070
      FILTER=0                                                           CHE02080
      IF(FRACTN .LE. (-0.989))FILTER=2                                   CHE02090
      IF((A+ARG*FRACTN) .LT. 0.0) FILTER=1                               CHE02100
      RETURN                                                             CHE02110
      END                                                                CHE02120
C                                                                        CHE02130
C     SUBROUTINE RADUS CALCULATES THE RADIUS TO THE FRONT SURFACE AT     CHE02140
C      THE SPECIFIED ANGLE AND ALSO THE DERIVATIVE WITH RESPECT TO       CHE02150
C      ANGLE.                                                            CHE02160
C                                                                        CHE02170
      SUBROUTINE RADUS(OSYMB,S,THETAP,RADD,RDGG)                         CHE02180
      DOUBLE PRECISION THETAP,OSYMB,S                                    CHE02190
      RADD=-OSYMB*S/(DSIN(THETAP)-S*DCOS(THETAP))                        CHE02200
      RDGG=S*OSYMB*(DCOS(THETAP)+S*DSIN(THETAP))/                        CHE02210
     1((DSIN(THETAP)-S*DCOS(THETAP))**2)                                 CHE02220
      RETURN                                                             CHE02230
      END                                                                CHE02240
```

```
C ***************************************************************** DETOOO10
C *                                                               * DETOOC20
C *                    D E T P L C T   F O R T R A N              * DETOOC30
C *                                                               * DETOOC40
C ***************************************************************** DETOOC50
C *                                                               * DETOOO60
C *       THIS PROGRAM IS DESIGNED TO COMPLETE THE RAY TRACE OF   * DETOOC70
C *    A MIRROR AND DETECTOR SYSTEM.   RAYS THAT LEAVE THE LENS   * DETOOO80
C *    ARE TRACED TO THE MIRROR, THEN REFLECTED OFF AND TRACED    * DETOOC90
C *    TO THE DETECTOR.   THIS PROCESS IS REPEATED FOR EACH       * DETOC100
C *    INCREMENT OF MIRROR ANGLE.   THE NUMBER OF RAYS STRIKING   * DETOC110
C *    THE UPPER AND LOWER HALF OF THE DETECTOR ARE RECORDED AT   * DETOC120
C *    EACH ANGLE AND THEN THIS INFORMATION IS OUTPUT SO THAT     * DETOC130
C *    APPROPRIATE PLOTS MAY BE MADE.                             * DETOO140
C *                                                               * DETOO150
C ***************************************************************** DETOC160
C                                                                   DETOO170
C   VARIABLE DEFINITIONS                                            DETOC180
C                                                                   DETOO190
C      BF      : DISTANCE FROM FOCAL POINT TO CONE APEX             DETOC200
C      ALPHA   : CONE ANGLE OF BACK LENS SURFACE                    DETOC210
C      ALFAP   : ANGLE OF RADIATION INCIDENT TO LENS                DETOO220
C      DETP    : DETECTOR POSITION                                  DETOC230
C      DETS    : DETECTOR RADIAL SIZE                               DETOC240
C      MIRP    : MIRROR PIVOT POSITION                              DETOC250
C      XC,YC,ZC    : COORDINATES OF RAY EXITING LENS                DETOC260
C      XM,YM,ZM    : COORDINATES OF RAY AT MIRROR SURFACE           DETOO270
C      YD,ZD       : COORDINATES OF RAY AT DETECTOR (XD=DETP)       DETOC280
C      CK,CL,CM    : DIRECTION COSINES OF RAY FROM LENS TO MIRROR   DETOC290
C      CKK,CLL,CMM: DIRECTION COSINES OF RAY FROM MIRROR TO DETECTOR DETOO300
C      COUNT1      : RAY COUNT ON UPPER DETECTOR                    DETOC310
C      COUNT2      : RAY COUNT ON LOWER DETECTOR                    DETOC320
C                                                                   DETOC330
C                                                                   DETOC340
C   SPECIFICATION STATEMENTS                                        DETOO350
C                                                                   DETOC360
      INTEGER J,K,L,OUT1,OUT2(500),OUT3(500),COUNT1(500),COUNT2(500), DETOC370
     1TERM1,TERM2                                                   DETOC380
      REAL BF,ALPHA,PI,PI2,DETP,DETS,MIRP,THETM(200),XO(500),YO(500), DETOC390
     1ZO(500),CK(500),CL(500),CM(500),D1,D2,XM(500),YM(500),ZM(500), DETOO400
     1CKK,CLL,CMM,YD(500),ZD(500),ALFAP,THETD(500)                  DETOC410
C                                                                   DETOC420
C   READ IN LENS CONSTANTS                                          DETOO430
C                                                                   DETOC440
      READ (5,100) BF,ALPHA,ALFAP                                   DETOO450
C                                                                   DETOO460
C                                                                   DETOC470
C                                                                   DETOC480
```

```
C    SET CONSTANTS                                                          DETOC490
C                                                                           DETOC500
       PI=3.14159                                                           DETOC510
       PI2=PI/2.                                                            DETOC520
       DETP=0.2                                                             DETOC530
       MIRP=1.4                                                             DETOC540
       THETC=66.0*PI/180                                                    DETOC550
       DETS=DETP*TAN(ALPHA)                                                 DETOC560
       THETM(1)=THETC-0.20404040                                            DETOC570
       J=0                                                                  DETOC580
       L=1                                                                  DETOC590
       OUT1=0                                                               DETOC600
       TERM1=0                                                              DETOC610
       TERM2=0                                                              DETOC620
C                                                                           DETOC630
C    READ IN RAY DATA REJECTING RAYS BLOCKED BY DETECTOR                    DETOC640
C                                                                           DETOC650
  10   J=J+1                                                                DETOC660
       READ(5,200) XO(J),YO(J),ZO(J),CK(J),CL(J),CM(J)                      DETOC670
       IF(SQRT(YO(J)**2+ZO(J)**2).LT.DETS) GO TO 11                         DETOC680
       IF(XO(J).EQ.1.0.AND.YO(J).EQ.1.0.AND.ZO(J).EQ.1.0) TERM1=J           DETOC690
       IF(XO(J).EQ.1.0.AND.YO(J).EQ.1.0.AND.ZO(J).EQ.1.0) GO TO 12          DETOC700
       GO TO 10                                                             DETOC710
  11   J=J-1                                                                DETOC720
       OUT1=OUT1+1                                                          DETOC730
       GO TO 10                                                             DETOC740
  12   CONTINUE                                                             DETOC750
C                                                                           DETOC760
C    INCREMENT MIRROR ANGLE AND CHECK FOR COMPLETED SWEEP                   DETOC770
C                                                                           DETOC780
  20   L=L+1                                                                DETOC790
       K=L-1                                                                DETOC800
       THETM(L)=THETM(K)+0.00404040                                         DETOC810
       THETD(L)=THETM(L)*180/PI                                             DETOC820
       IF(THETM(L).GT.(THETC+0.20)) TERM2=L-1                               DETOC830
C      WRITE(6,2000) THETM(L),OUT1,TERM1,TERM2                              DETOC840
       IF(THETM(L).GT.(THETC+0.20)) GO TO 40                                DETOC850
       OUT2(L)=0                                                            DETOC860
       OUT3(L)=0                                                            DETOC870
       COUNT1(L)=0                                                          DETOC880
       COUNT2(L)=0                                                          DETOC890
C                                                                           DETOC900
C                                                                           DETOC910
C    RAY TRACE FROM BACK LENS SURFACE TO MIRROR AND THEN TO DETECTOR        DETOC920
C                                                                           DETOC930
C                                                                           DETOC940
       J=0                                                                  DETOC950
  30   J=J+1                                                                DETOC960
```

```
      IF(J.EC.TERM1) GC TC 20                                                    DETOC970
C                                                                                DET00980
C     FIND MIRRCR INTERSECTION POINT  DISCARD RAYS CUTSIDE MIRRCR                DET00990
C                                                                                DET01000
      D1=(TAN(THETM(L))*(XO(J)-MIRP)-YO(J))/(CL(J)-CK(J)*TAN(THETM(L)))          DET01010
      XM(J)=D1*CK(J)+XO(J)                                                       DET01020
      YM(J)=D1*CL(J)+YO(J)                                                       DET01030
      ZM(J)=D1*CM(J)+ZO(J)                                                       DET01040
C     WRITE(6,1000) XM(J),YM(J),ZM(J)                                            DET01050
      IF(SQRT(ZM(J)**2+(COS(THETM(L))*YM(J))**2).GT.1.0)                         DET01060
     1OUT2(L)=OUT2(L)+1                                                          DET01070
      IF(SQRT(ZM(J)**2+(COS(THETM(L))*YM(J))**2).GT.1.0) GO TO 30                DET01080
C                                                                                DET01090
C     FIND DIRECTION OF RAY REFLECTED FROM MIRRCR SURFACE                        DET01100
C                                                                                DET01110
      CKK=+COS(2*THETM(L))*CK(J)+SIN(2*THETM(L))*CL(J)                           DET01120
      CLL=-COS(2*THETM(L))*CL(J)+SIN(2*THETM(L))*CK(J)                           DET01130
      CMM=CM(J)                                                                  DET01140
C     WRITE(6,1010) CK(J),CL(J),CM(J),CKK,CLL,CMM                                DET01150
C                                                                                DET01160
C     FIND INTERSECTION WITH DETECTOR  DISCARD RAYS MISSING DETECTOR             DET01170
C                                                                                DET01180
      D2=(DETF-XM(J))/CKK                                                        DET01190
      YD(J)=D2*CLL+YM(J)                                                         DET01200
      ZD(J)=D2*CMM+ZM(J)                                                         DET01210
      IF(SQRT(YD(J)**2+ZD(J)**2).GT.DETS) OUT3(L)=OUT3(L)+1                      DET01220
      IF(SQRT(YD(J)**2+ZD(J)**2).GT.DETS) GO TO 30                               DET01230
C                                                                                DET01240
C     COUNT RAYS HITTING UPPER AND LOWER PORTIONS OF DETECTOR                    DET01250
C                                                                                DET01260
      IF(YD(J).GT.0.0) COUNT1(L)=COUNT1(L)+1                                     DET01270
      IF(YD(J).LT.0.0) COUNT2(L)=COUNT2(L)+1                                     DET01280
      GO TO 30                                                                   DET01290
C                                                                                DET01300
C     WRITE OUTPUT                                                               DET01310
C                                                                                DET01320
   40 CONTINUE                                                                   DET01330
C     WRITE(6,2005) ALFAP                                                        DET01340
      DO 50 L=2,TERM2                                                            DET01350
      WRITE(6,2010) THETD(L),COUNT1(L),COUNT2(L)                                 DET01360
   50 CONTINUE                                                                   DET01370
      STOP                                                                       DET01380
C                                                                                DET01390
C     FORMAT STATEMENTS                                                          DET01400
C                                                                                DET01410
  100 FORMAT(1X,3F10.7)                                                          DET01420
  200 FORMAT(1X,6F10.7)                                                          DET01430
C100C FORMAT(1X,3F10.6)                                                          DET01440
```

```
C101C FORMAT(1X,6F10.6)                                          DET01450
C200C FORMAT(1X,F10.7,3I10)                                      DET01460
C200S FORMAT(1X,F10.6)                                           DET01470
  201C FORMAT(1X,F10.6,2I5)                                      DET01480
C                                                                DET01490
      END                                                        DET01500
```

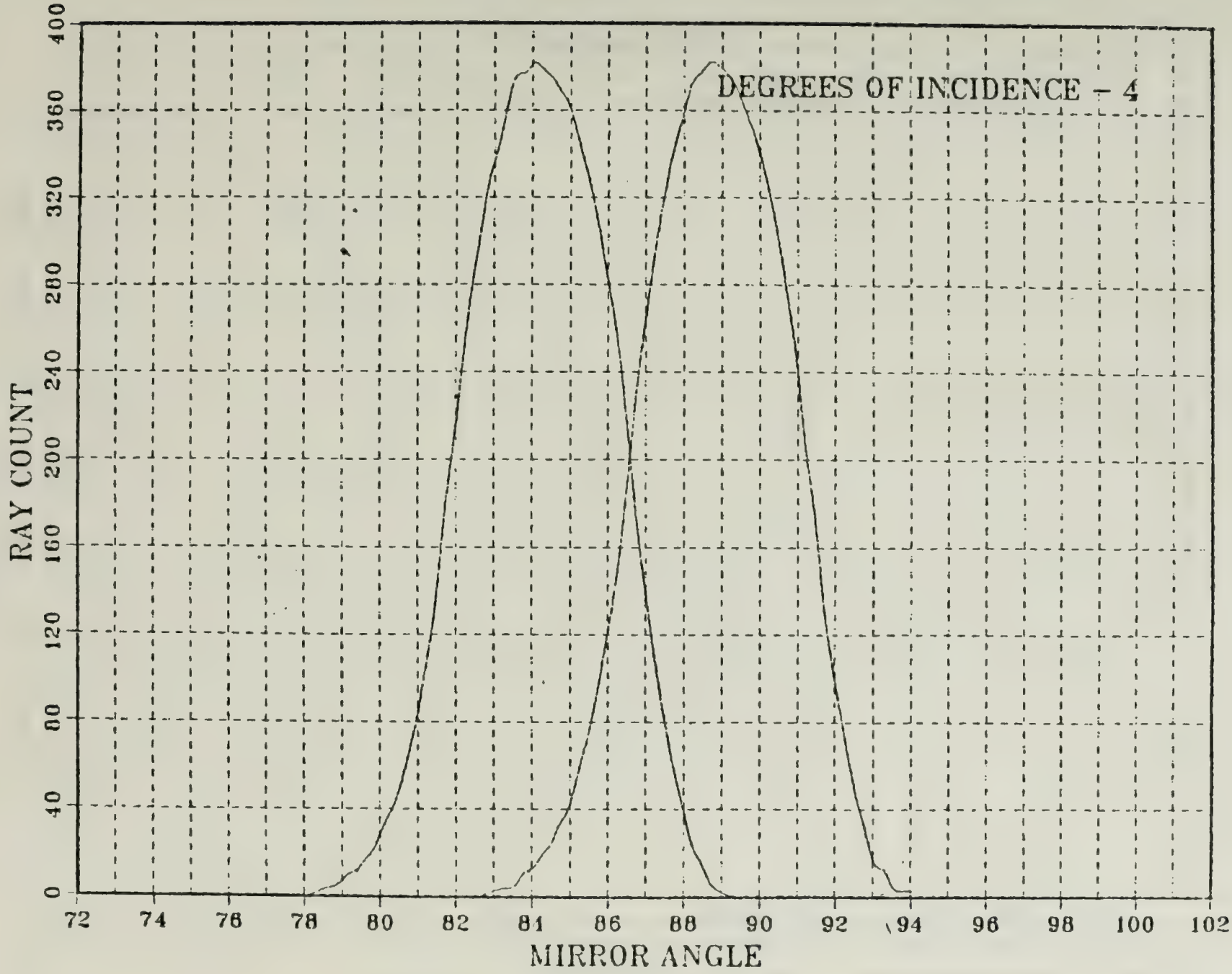

Figure D-1. Detector Signal at 4 Degrees Incidence.

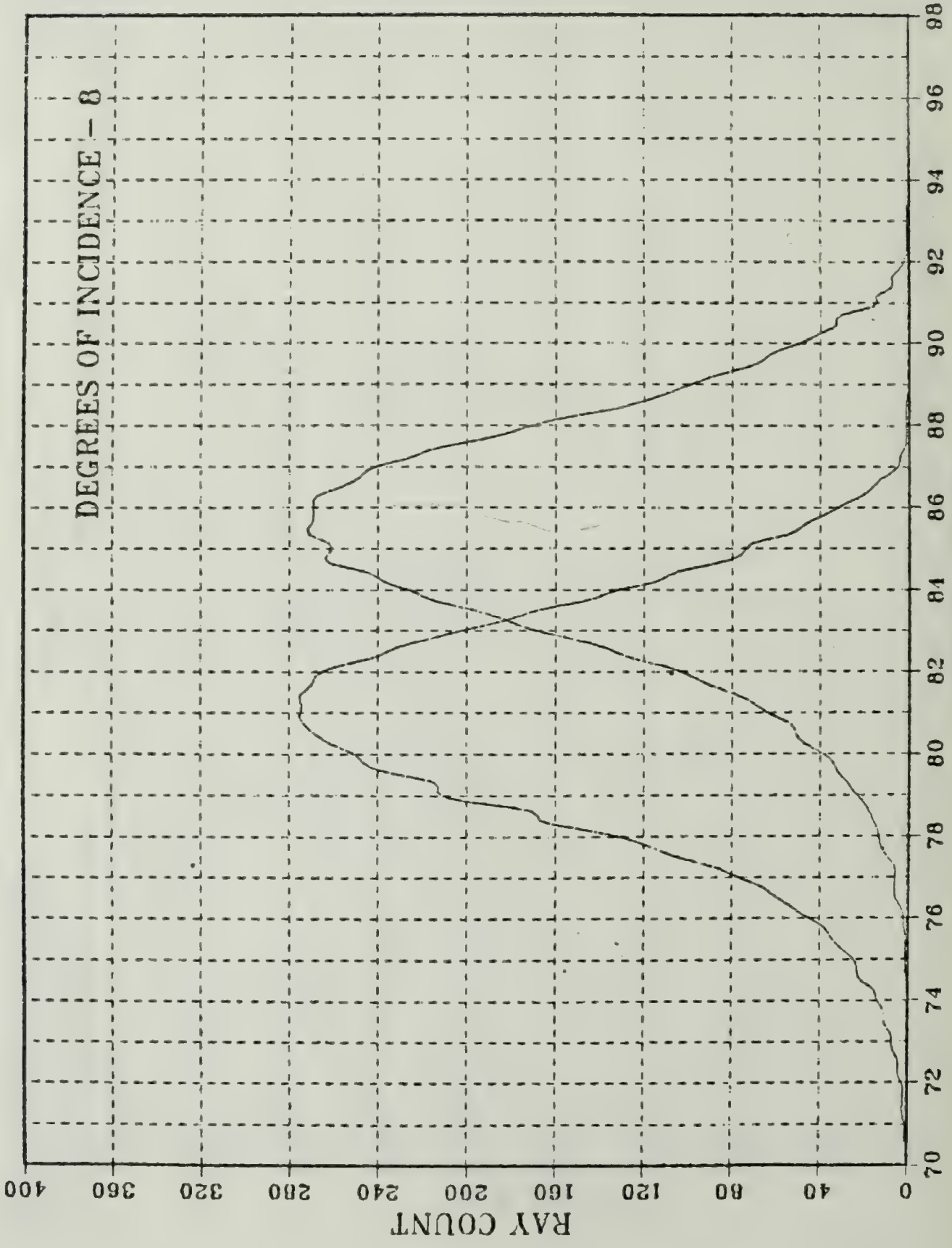

Figure D-2. Detector Signal at 8 Degrees Incidence.

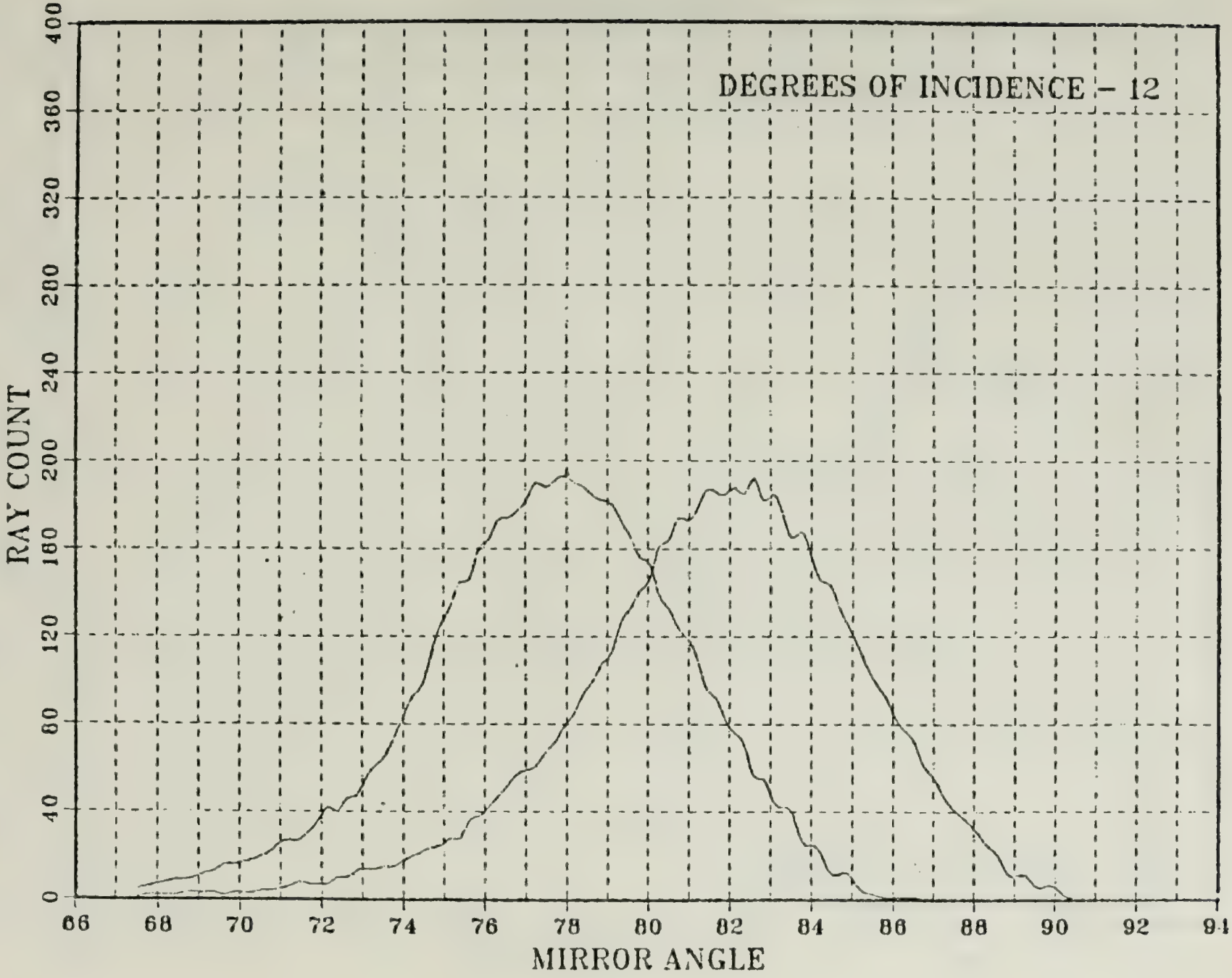

Figure D-3. Detector Signal at 12 Degrees Incidence.

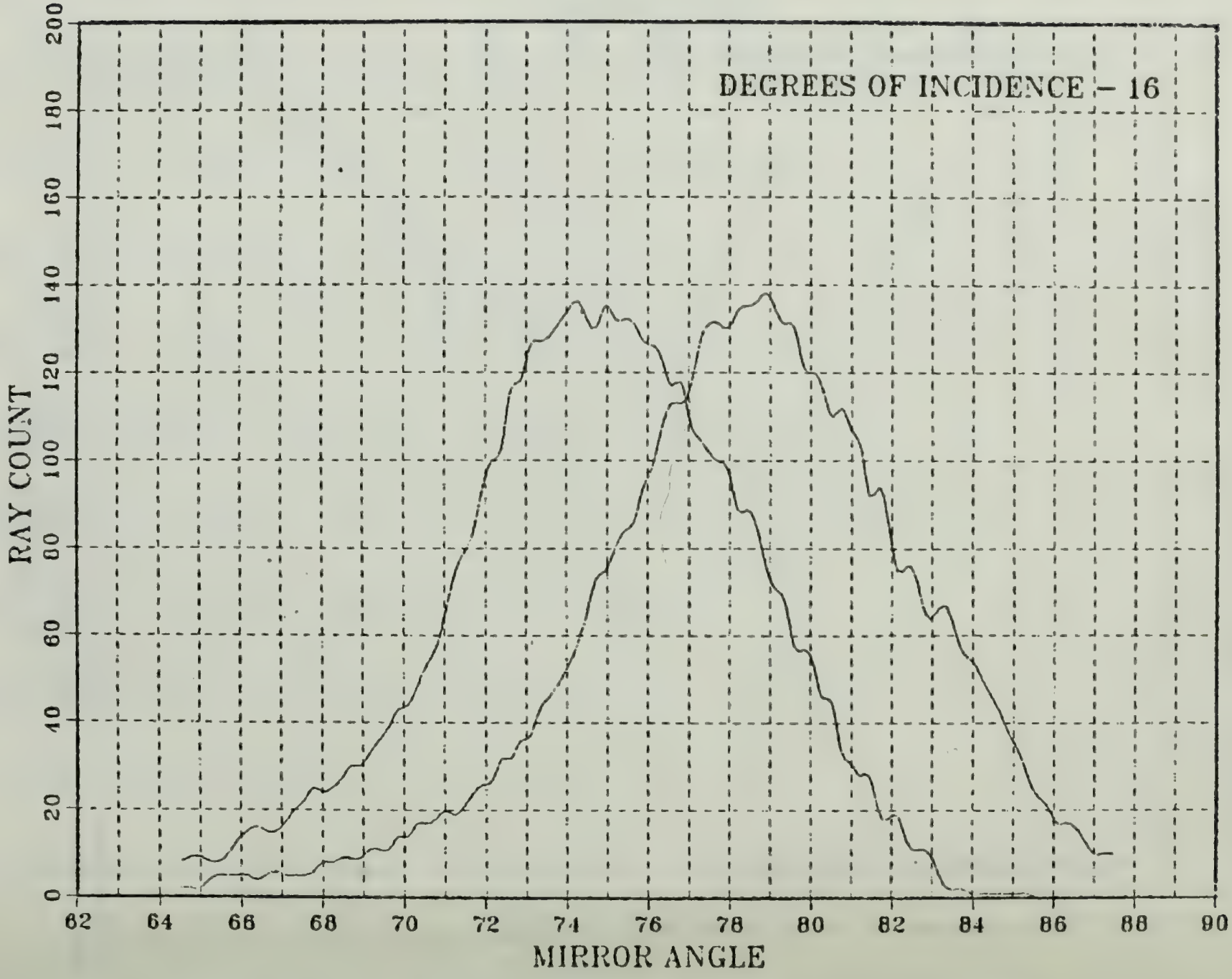

Figure D-4. Detector Signal at 16 Degrees Incidence.

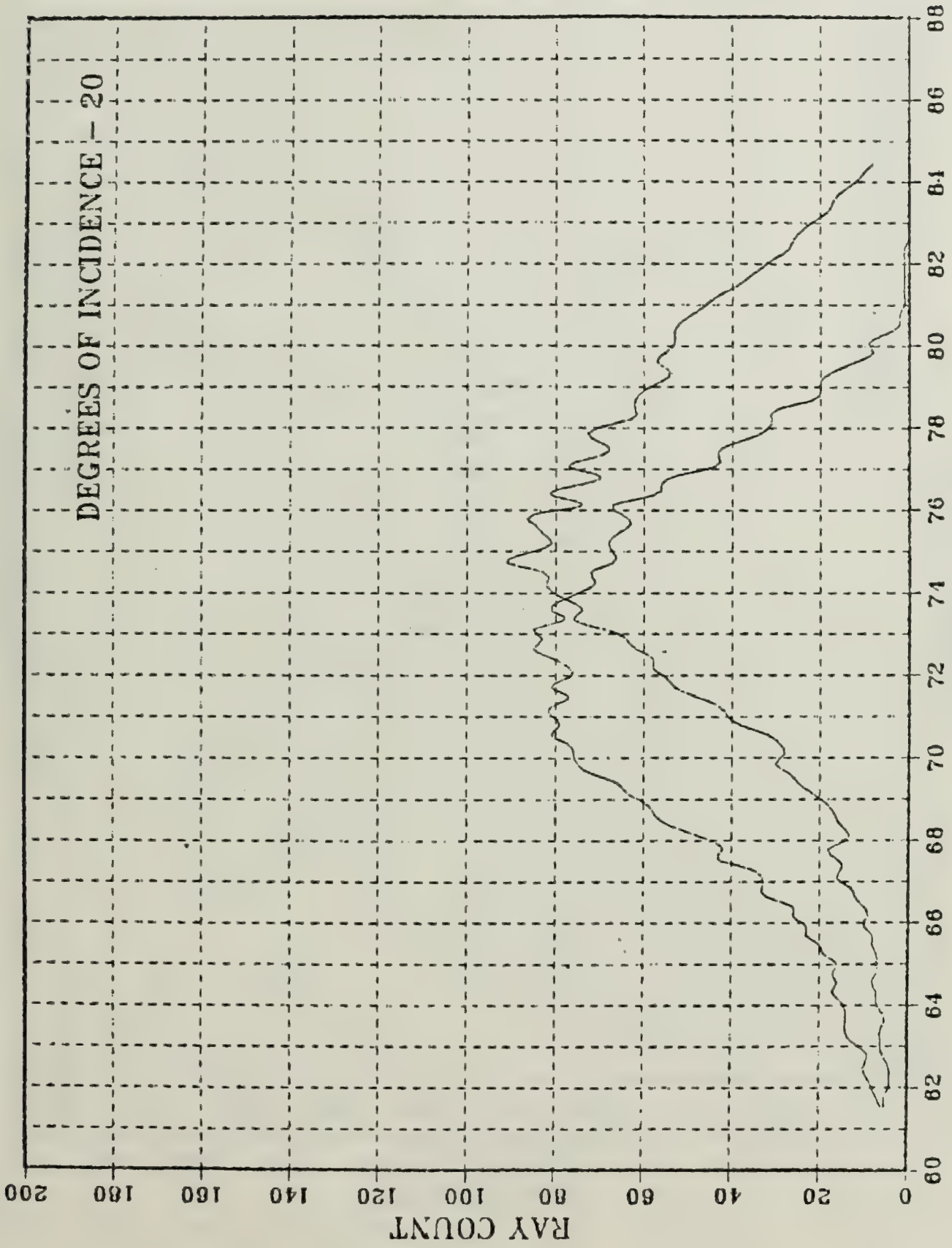

Figure D-5. Detector Signal at 20 Degrees Incidence.

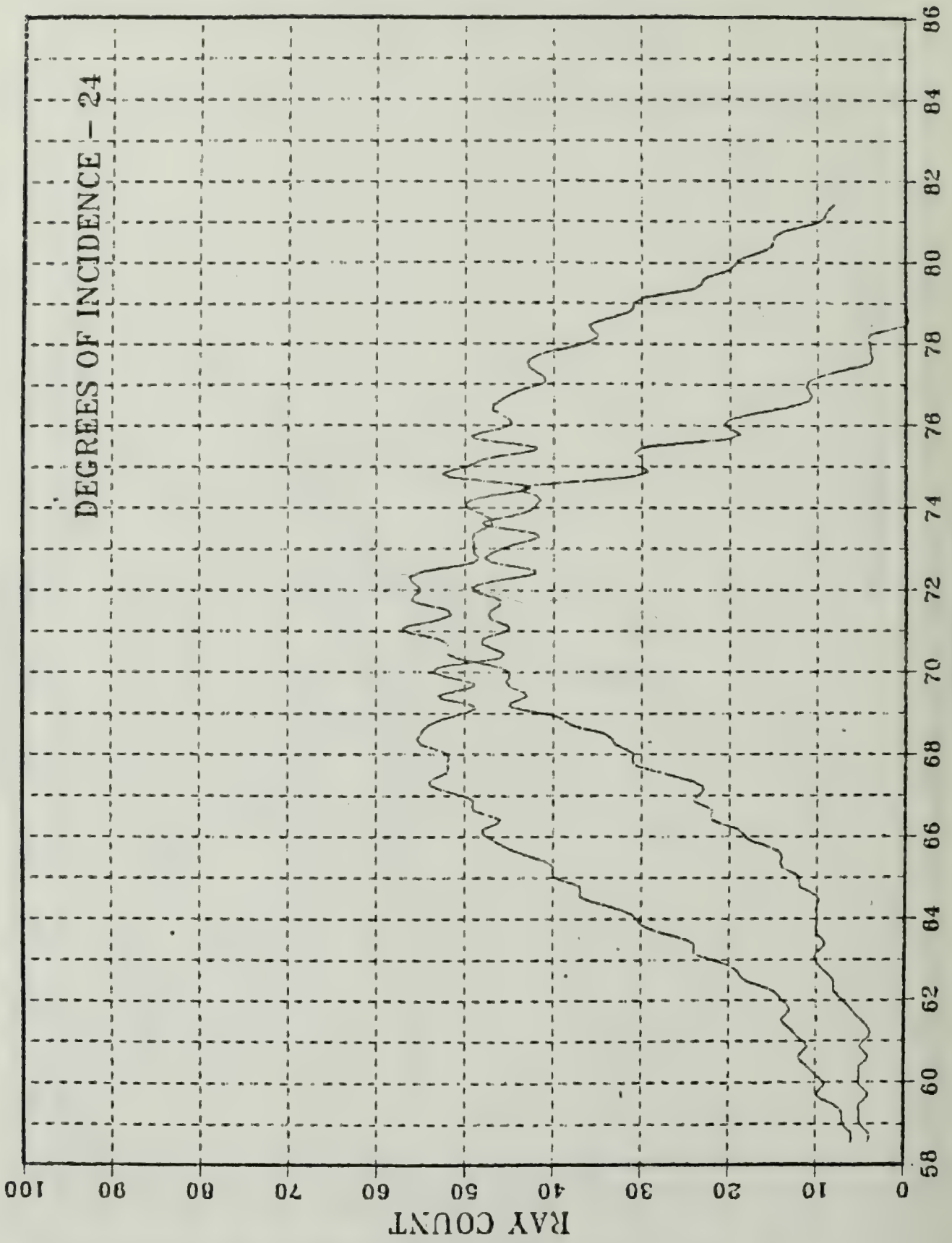

Figure D-6. Detector Signal at 24 Degrees Incidence.

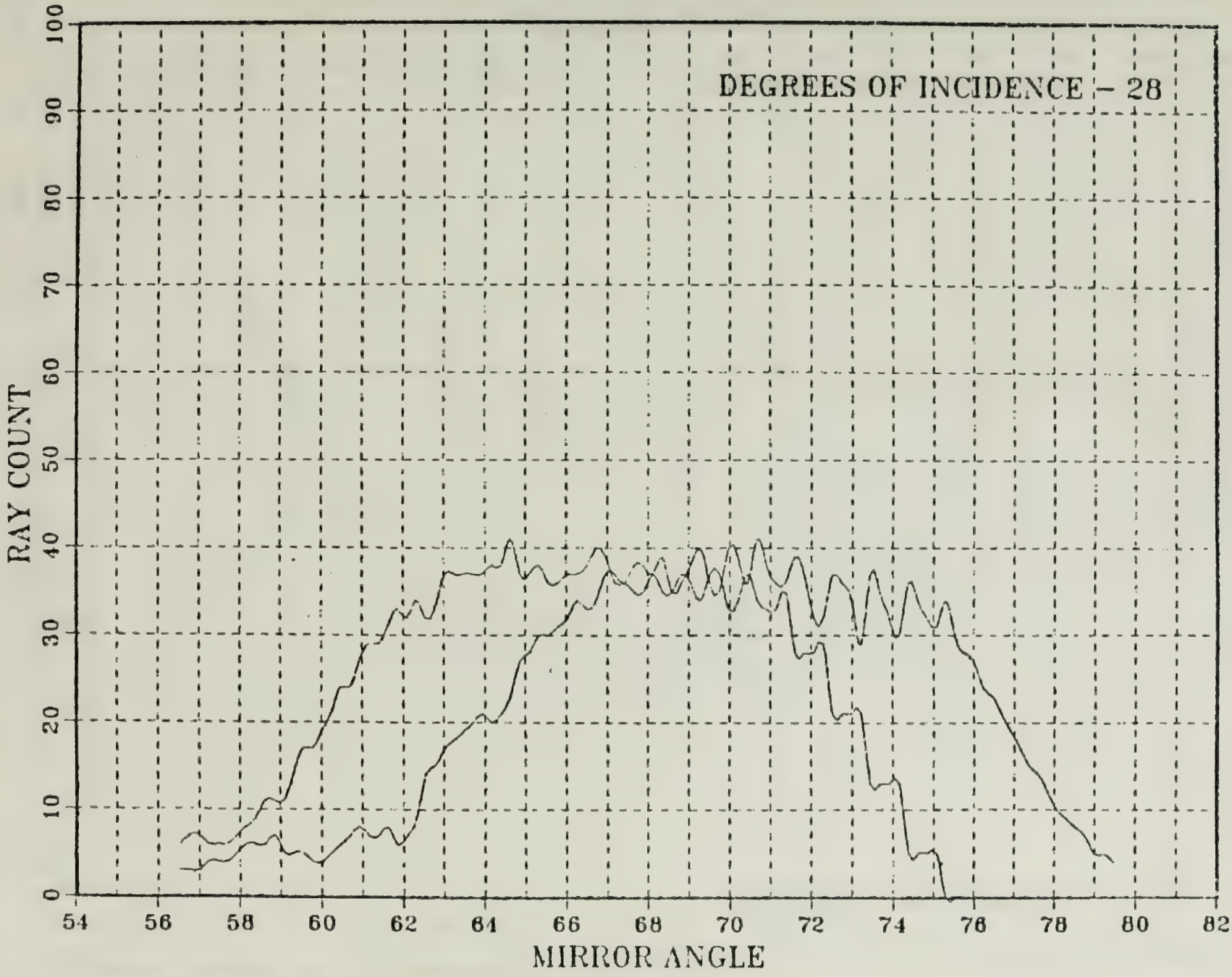

Figure D-7. Detector Signal at 28 Degrees Incidence.

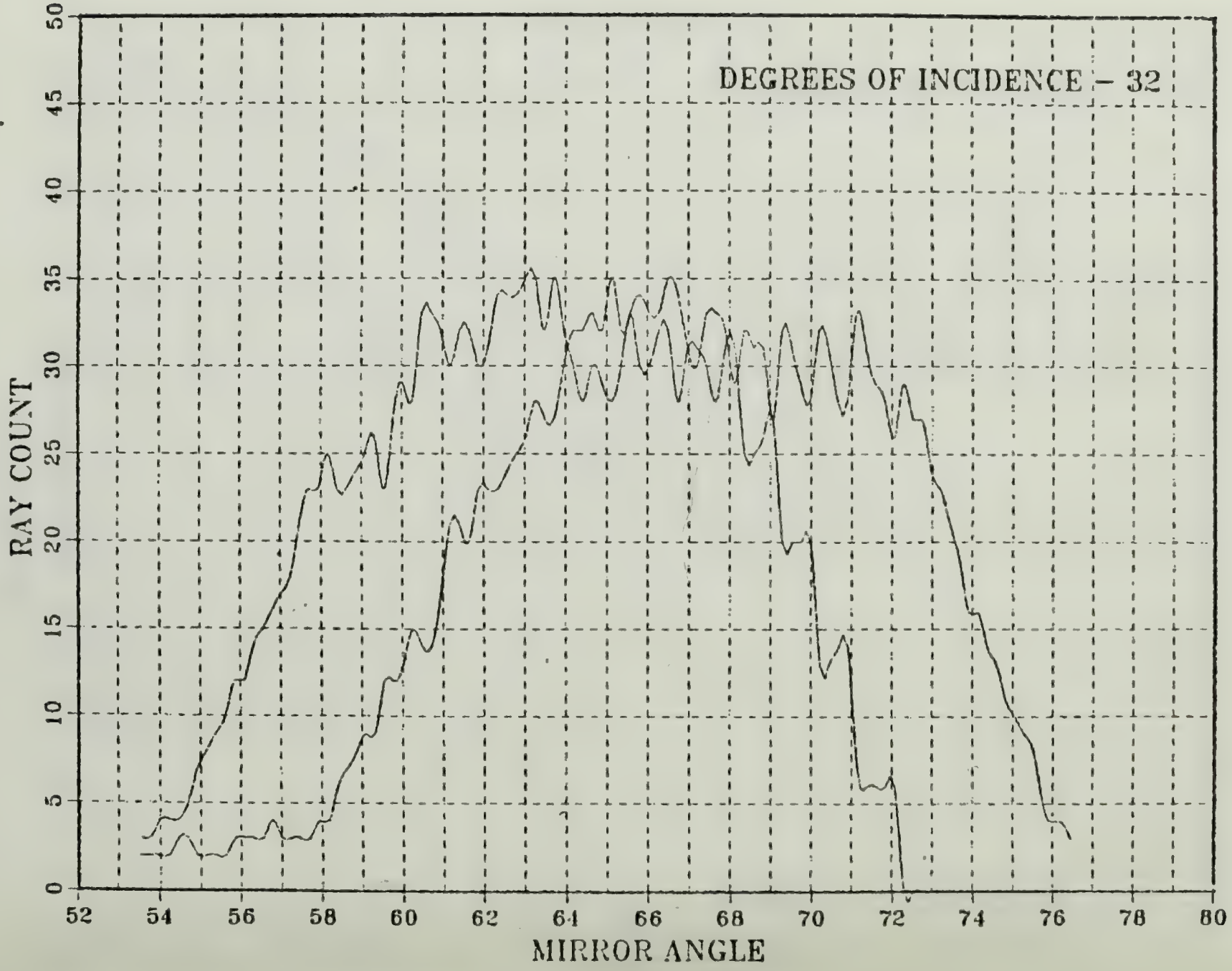

Figure D-8. Detector Signal at 32 Degrees Incidence.

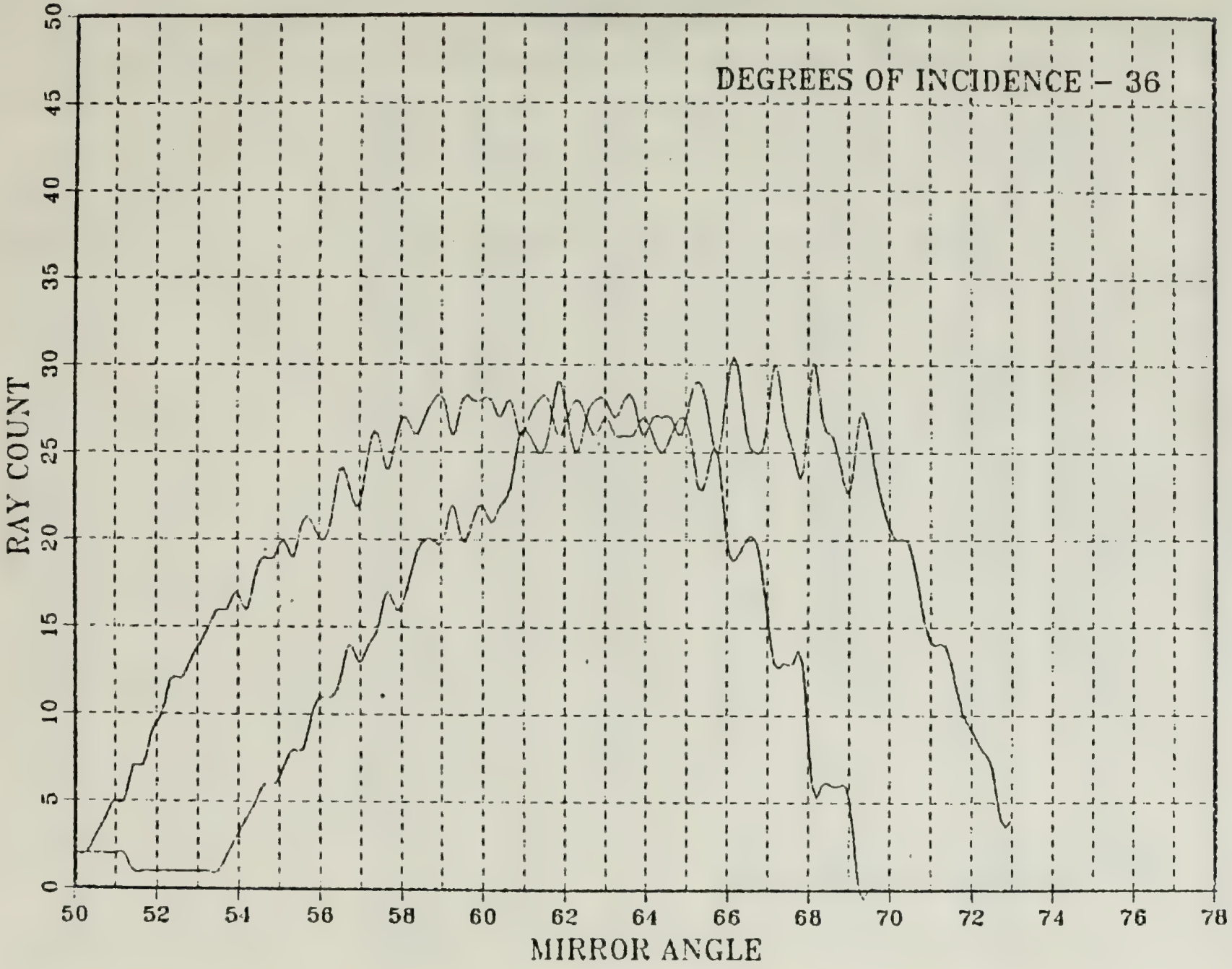

Figure D-9. Detector Signal at 36 Degrees Incidence.

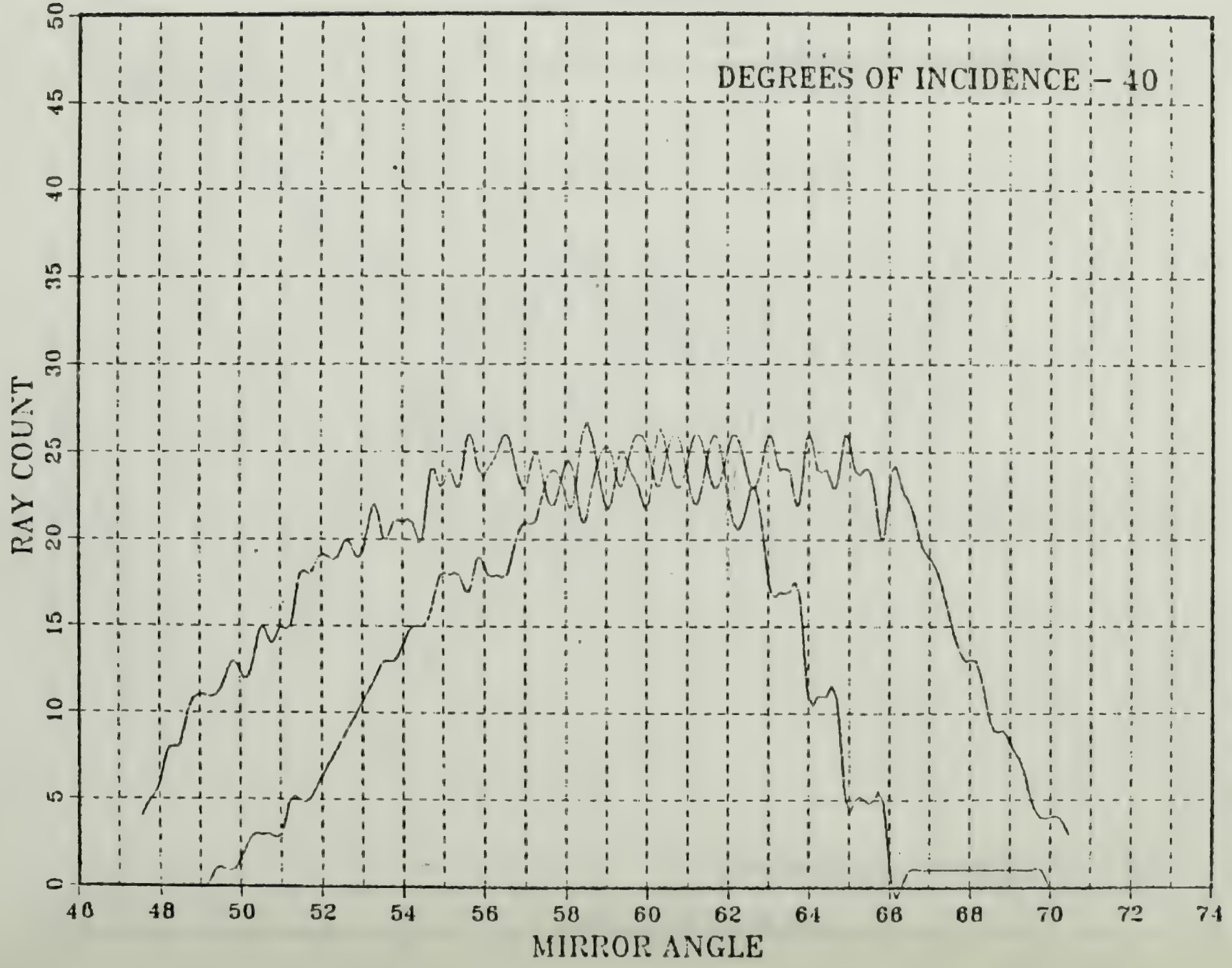

Figure D-10. Detector Signal at 40 Degrees Incidence.

LIST OF REFERENCES

1. Frazier, Robert L., Exterior Ballistics, Guidance Laws and Optics for a Gun-Launched Missile, M. S. Thesis, Naval Postgraduate School, Monterey, California, December 1980.

2. Terrell, James M., Conical Lens for a 5"/54 Gun Launched Missile, M. S. Thesis, Naval Postgraduate School, Monterey, California, June 1981.

3. Amichai, Oded, Sharp Nose Lens Design Using Refractive Index Gradient, Naval Postgraduate School Contractor Report NPS67-82-003CR, Monterey, California, June 1982.

4. Carr, Herbert M., Aerodynamically Efficient Gradient Refractive Index Missile Seeker Lens, M. S. Thesis, Naval Postgraduate School, Monterey, California, October 1982.

5. Marchand, Erich W., Gradient Index Optics, Academic Press, 1978.

6. U. S. Army Missile Command Technical Report RG-CR-82-6, Gradient Index Lens Research-Final Report, by D. T. Moore, p. 36, 19 October 1981.

7. U. S. Army Missile Command Technical Report H8 3-3-BO20-1, Gradient Index Optics Application by R. L. Light, 30 September 1983.

8. D. S. Davidson and A. E. Fuhs, Tracing Nearly Radial Lines in a Spherically Symmetric Gradient Index Lens, Applied Optics, Letter to Editor (in press).

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Directorate of Land Armament and Electronics Engineering and Maintenance Attn: Colonel B. L. Code National Defense Headquarters Ottawa, Ontario, Canada K1A 0K2	2
4.	Distinguished Professor A. E. Fuhs Code 67Fu Department of Aeronautics Naval Postgraduate School Monterey, California 93943	4
5.	Col. R. Larriva, USMC Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209	2
6.	Charles R. Christensen Research Directorate U. S. Army Missile Command (DRSMI-RRO) Redstone Arsenal, Alabama 35898	1
7.	Commander, U. S. Army Missile Command (DRSMI-ROA) Redstone Arsenal, Alabama 35898 Attn: CAPT H. Carr	1
8.	Professor A. E. Milne Code 61Mn Naval Postgraduate School Monterey California 93943	1
9.	Professor D. T. Moore University of Rochester Rochester, New York 14627	1

10. Dr. Lloyd Smith — 1
Code 3205
Naval Weapons Center
China Lake, California 93555

11. Directorate of Land Armament and Electronics Engineering and Maintenance — 2
Attn: Capt. D. Davidson
National Defense Headquarters
Ottawa, Ontario, Canada K1A OK2

12. Directorate of Personnel Education and Development (DPED) — 1
National Defense Headquarters
Ottawa, Ontario, Canada K1A OK2

1 3 37

210152

Thesis
D1639 Davidson
c.1 Supersonic missile seeker lens system using gradient index material for first element lens.

210152

Thesis
D1639 Davidson
c.1 Supersonic missile seeker lens system using gradient index material for first element lens.

Supersonic missile seeker lens system us

3 2768 002 09566 3
DUDLEY KNOX LIBRARY